

**KLASIFIKASI PENYAKIT KULIT SAPI MENGGUNAKAN METODE CNN
(CONVOLUTIONAL NEURAL NETWORK)**

SKRIPSI

Oleh :
ARKHA YUSFINA PRAMITHA
NIM. 19650053



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2026**

**KLASIFIKASI PENYAKIT KULIT SAPI MENGGUNAKAN METODE
CNN (*CONVOLUTIONAL NEURAL NETWORK*)**

SKRIPSI

Diajukan Kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh:
ARKHA YUSFINA PRAMITHA
NIM. 19650053

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2026**

HALAMAN PERSETUJUAN

**KLASIFIKASI PENYAKIT KULIT SAPI MENGGUNAKAN METODE
CNN (*CONVOLUTIONAL NEURAL NETWORK*)**

SKRIPSI

**Oleh:
ARKHA YUSFINA PRAMITHA
NIM. 19650053**

**Telah Diperiksa dan Disetujui untuk Diuji
Tanggal: 18 Juni 2026**

Pembimbing I



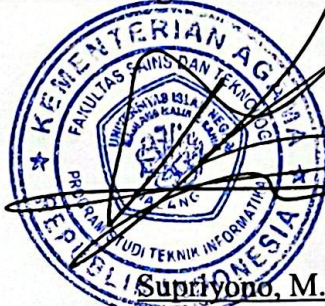
Dr. Ir. Fachrul Kurniawan, M.MT., IPU
NIP. 19771020 200912 1 001

Pembimbing II



Shoffin Nahwa Utama, M.T
NIP. 19860703 202012 1 003

**Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang**



Supriyono, M.Kom
NIP. 19841010 201903 1 012

HALAMAN PENGESAHAN

KLASIFIKASI PENYAKIT KULIT SAPI MENGGUNAKAN METODE CNN (*CONVOLUTIONAL NEURAL NETWORK*)

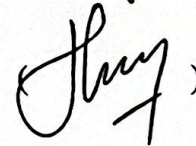
SKRIPSI

Oleh:
ARKHA YUSFINA PRAMITHA
NIM. 19650053

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 23 Juni 2026

Susunan Dewan Penguji

Ketua Penguji	: <u>A'la Syauqi, M.Kom</u> NIP. 19771201 200801 1 007
Anggota Penguji I	: <u>Supriyono, M.Kom</u> NIP. 19841010 201903 1 012
Anggota Penguji II	: <u>Dr. Ir. Fachrul Kurniawan, M.MT., IPU</u> NIP. 19771020 200912 1 001
Anggota Penguji III	: <u>Shoffin Nahwa Utama, M.T</u> NIP. 19860703 202012 1 003



Mengetahui dan Mengesahkan,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Supriyono, M.Kom
NIP. 19841010 201903 1 012

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Arkha Yusfina Pramitha
NIM : 19650053
Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika
Judul Skripsi : Klasifikasi Penyakit Kulit Sapi Menggunakan Metode CNN (*Convolutional Neural Network*)

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 26 Juni 2026

Yang membuat pernyataan,



Arkha Yusfina Pramitha

NIM. 19650053

MOTTO

“Boleh jadi kamu membenci sesuatu, padahal itu baik bagimu dan boleh jadi kamu menyukai sesuatu, padahal itu buruk bagimu. Allah mengetahui, sedangkan kamu tidak mengetahui.”

(QS. Al-Baqarah: 216)

HALAMAN PERSEMBAHAN

Dengan mengucapkan rasa syukur kepada Allah SWT yang telah memberikan rahmat, kesehatan, kekuatan, dan kemudahan sehingga penulis dapat menyelesaikan skripsi ini. Dengan segenap hati, hasil karya ini penulis persembahkan kepada:

1. Kedua orang tua tercinta, terima kasih atas segala cinta yang tidak pernah berkurang, doa yang tidak pernah putus, serta pengorbanan yang tidak pernah terhitung nilainya. Semoga karya ini dapat menjadi salah satu bentuk kebahagiaan dan kebanggaan untuk Ayah dan Ibu.
2. Untuk adik tercinta, terima kasih atas dukungan, perhatian, dan semangat yang selalu diberikan, baik secara langsung maupun tidak langsung.
3. Untuk dosen pembimbing dan seluruh dosen, terima kasih atas ilmu, bimbingan, arahan, serta kesabaran yang telah diberikan selama masa perkuliahan hingga proses penyusunan skripsi ini.
4. Untuk sahabat dan teman-teman seperjuangan, terima kasih atas kebersamaan, dukungan, bantuan, dan kenangan selama menempuh pendidikan. Terima kasih telah menjadi bagian dari proses yang tidak akan pernah penulis lupakan.
5. Untuk diri sendiri, terima kasih telah bertahan sejauh ini. Terima kasih karena tetap berjalan ketika keadaan terasa berat, ketika keraguan menghampiri, dan ketika banyak hal tidak sesuai dengan harapan.

Akhir kata, karya ini penulis persembahkan untuk semua orang yang pernah hadir dalam perjalanan hidup penulis, yang mendoakan, membantu, mendukung, mengingatkan, dan memberikan pelajaran berharga. Semoga Allah SWT membalas setiap kebaikan yang telah diberikan dengan balasan yang berlipat ganda.

“Mungkin saya tidak sampai di garis akhir secepat yang lain. Mungkin saya membutuhkan waktu yang lebih panjang dan menghadapi lebih banyak rintangan dibandingkan sebagian orang. Namun perjalanan ini mengajarkan bahwa setiap orang memiliki waktunya masing-masing. Terlambat bukan berarti gagal, dan berjalan dengan langkah yang berbeda bukan berarti tertinggal.”

Terima kasih untuk setiap doa, pengorbanan, dukungan, dan cinta yang mengantarkan hingga titik ini.

KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah melimpahkan segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Klasifikasi Penyakit Kulit Sapi Menggunakan Metode CNN (*Convolutional Neural Network*)” dengan baik.

Penyusunan skripsi ini tidak terlepas dari bantuan, bimbingan, dukungan, dan doa dari berbagai pihak. Oleh karena itu, penulis menyampaikan ucapan terima kasih kepada:

1. Prof. Dr. Hj. Ilfi Nur Diana, M.Si, CAHRM, CRMP selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Agus Mulyono, M.Kes selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Supriyono, M.Kom selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Dr. Ir. Fachrul Kurniawan, M.MT., IPU selaku Dosen Pembimbing I dan Bapak Shoffin Nahwa Utama, M.T selaku Dosen Pembimbing II atas bimbingan, arahan, dan motivasi yang sangat berarti selama penyusunan skripsi ini.
5. Bapak A’la Syauqi, M.Kom selaku Dosen Penguji I dan Bapak Supriyono, M.Kom selaku Dosen Penguji II atas masukan dan saran yang membangun.
6. Dr. M. Ainul Yaqin, M.Kom selaku Dosen Wali atas arahan dan bimbingannya selama masa studi.

7. Segenap dosen dan staff akademik Program Studi Informatika yang telah memberikan ilmu dan bantuan yang selama masa studi.
8. Ayah, Ibu, dan Adik tercinta yang telah memberikan dukungan, doa, dan motivasi sehingga penulis dapat menyelesaikan penyusunan skripsi ini.
9. Teman-teman angkatan 2019 Program Studi Teknik Informatika yang selalu memberikan semangat, bantuan, dan motivasi selama masa studi.
10. Semua pihak yang tidak dapat disebutkan satu per satu yang telah memberikan doa, dukungan, dan bantuan dalam penyelesaian skripsi ini.
11. Penulis sendiri yang telah berusaha dan bertanggung jawab dalam menyelesaikan skripsi ini.

Penulis menyadari skripsi ini masih terdapat banyak kekurangan. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan dan pengembangan penelitian di masa mendatang.

Akhir kata, penulis berharap semoga skripsi ini dapat memberikan manfaat bagi pembaca serta menjadi tambahan pengetahuan, khususnya dalam bidang pengolahan citra digital dan kecerdasan buatan.

Malang, 26 Juni 2026

Penulis

DAFTAR ISI

HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN.....	iv
PERNYATAAN KEASLIAN TULISAN.....	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR.....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL	xiv
ABSTRAK	xv
ABSTRACT	xvi
مستخلص البحث	xvii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	6
1.3 Tujuan Penelitian	6
1.4 Batasan Masalah	6
1.5 Manfaat Penelitian	7
BAB II STUDI PUSTAKA.....	8
2.1 Penelitian Terkait	8
2.2 Sapi	9
2.3 <i>Lumpy Skin Disease</i> (LSD).....	10
2.4 <i>Convolutional Neural Network</i> (CNN).....	11
2.5 Penerapan CNN Pada Klasifikasi Citra Penyakit	13
2.6 Arsitektur VGG16 dan VGG19	15
2.7 Evaluasi Kinerja Model	17
BAB III DESAIN DAN IMPLEMENTASI	20
3.1 Desain Penelitian	20
3.2 Pengumpulan Data	21
3.3 Desain Sistem.....	23
3.4 <i>Image Preprocessing</i>	25
3.5 Implementasi Metode.....	28
3.6 Proses Training	34
3.7 Skenario Uji Coba.....	36
3.8 Evaluasi Model	38
BAB IV HASIL DAN PEMBAHASAN	40
4.1 Uji Coba	40
4.2 Hasil Uji Coba.....	41
4.2.1 Uji Coba Skenario 1A.....	42
4.2.2 Uji Coba Skenario 1B	45
4.2.3 Uji Coba Skenario 2A.....	49
4.2.4 Uji Coba Skenario 2B	53
4.2.5 Uji Coba Skenario 3A.....	57

4.2.6	Uji Coba Skenario 3B	61
4.2.7	Hasil Uji Coba Skenario Terbaik	65
4.3	Implementasi dan Pengujian Sistem Berbasis Web	66
4.3.1	Implementasi Antarmuka Sistem	66
4.3.2	Pengujian Sistem	69
4.4	Pembahasan	72
4.4.1	Pengaruh Proporsi Data	74
4.4.2	Perbandingan Arsitektur VGG16 dan VGG19	76
4.5	Integrasi Islam	78
BAB V KESIMPULAN DAN SARAN		81
5.1	Kesimpulan	81
5.2	Saran	82
DAFTAR PUSTAKA		84

DAFTAR GAMBAR

Gambar 2.1 Struktur Arsitektur VGG16.....	16
Gambar 2.2 Struktur Arsitektur VGG19.....	17
Gambar 3.1 Tahapan Penelitian.....	20
Gambar 3.2 Desain Sistem.....	24
Gambar 3.3 Proses Pengolahan Data.....	25
Gambar 3.4 <i>Rotate</i>	26
Gambar 3.5 <i>Zoom</i>	27
Gambar 3.6 Vertical Flip	27
Gambar 3.7 Horizontal Flip	27
Gambar 3.8 Arsitektur CNN VGG16 dan VGG19 (Sumber: Bilal, 2022).....	28
Gambar 3.9 Perhitungan Proses Konvolusi	30
Gambar 3.10 Penerapan Fungsi Aktivasi ReLU.....	30
Gambar 3.11 Penerapan <i>Max Pooling</i>	31
Gambar 3.12 Penerapan <i>Flattening</i>	32
Gambar 4.1 Kurva <i>Accuracy</i> dan <i>Loss</i> Skenario 1A	42
Gambar 4.2 <i>Confusion Matrix</i> Validasi Skenario 1A	43
Gambar 4.3 <i>Confusion Matrix</i> Uji Skenario 1A	44
Gambar 4.4 Kurva <i>Accuracy</i> dan <i>Loss</i> Skenario 1B.....	46
Gambar 4.5 <i>Confusion Matrix</i> Validasi Skenario 1B	47
Gambar 4.6 <i>Confusion Matrix</i> Uji Skenario 1B	48
Gambar 4.7 Kurva <i>Accuracy</i> dan <i>Loss</i> Skenario 2A	49
Gambar 4.8 <i>Confusion Matrix</i> Validasi Skenario 2A	51
Gambar 4.9 <i>Confusion Matrix</i> Uji Skenario 2A	52
Gambar 4.10 Kurva <i>Accuracy</i> dan <i>Loss</i> Skenario 2B.....	53
Gambar 4.11 <i>Confusion Matrix</i> Validasi Skenario 2B	55
Gambar 4.12 <i>Confusion Matrix</i> Uji Skenario 2B	56
Gambar 4.13 Kurva <i>Accuracy</i> dan <i>Loss</i> Skenario 3A	57
Gambar 4.14 <i>Confusion Matrix</i> Validasi Skenario 3A.....	59
Gambar 4.15 <i>Confusion Matrix</i> Uji Skenario 3A	60
Gambar 4.16 Kurva <i>Accuracy</i> dan <i>Loss</i> Skenario 3B.....	61
Gambar 4.17 <i>Confusion Matrix</i> Validasi Skenario 3B	63
Gambar 4.18 <i>Confusion Matrix</i> Uji Skenario 3B	64
Gambar 4.19 Tampilan sistem dengan hasil prediksi kelas Normal Skin	67
Gambar 4.20 Tampilan sistem dengan hasil prediksi kelas Lumpy Skin	68
Gambar 4.21 Perbandingan Akurasi Data Validasi dan Uji Seluruh Skenario.....	74

DAFTAR TABEL

Tabel 2.1 Penelitian Terkait	8
Tabel 2.2 Confusion Matrix	17
Tabel 3.1 Beberapa Contoh Dataset Penyakit Kulit Sapi	22
Tabel 3.2 Skenario Pengujian Dengan Variasi Dataset	37
Tabel 3.3 Skenario Pengujian Dengan Variasi Arsitektur	37
Tabel 3.4 Kombinasi Skenario Uji Coba	37
Tabel 4.1 Nilai Parameter	41
Tabel 4.2 Skenario Uji Coba.....	41
Tabel 4.3 Perbandingan Akurasi Data Uji Seluruh Skenario	65
Tabel 4.4 Hasil Pengujian Sistem	70
Tabel 4.5 Rekapitulasi Hasil Data Validasi Seluruh Skenario.....	73
Tabel 4.6 Rekapitulasi Hasil Data Uji Seluruh Skenario.....	73

ABSTRAK

Pramitha, Arkha Yufina. 2026. **Klasifikasi Penyakit Kulit Sapi Menggunakan Metode CNN (*Convolutional Neural Network*)**. Skripsi. Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Ir. Fachrul Kurniawan, M.MT., IPU (II) Shoffin Nahwa Utama, M.T

Kata Kunci: Klasifikasi Citra, *Convolutional Neural Network*, *Lumpy Skin Disease*, VGG16, VGG19, Gradio

Lumpy Skin Disease (LSD) merupakan penyakit kulit menular pada sapi yang disebabkan oleh virus dan dapat menimbulkan kerugian ekonomi signifikan bagi peternak. Deteksi dini penyakit ini secara visual masih bergantung pada keahlian tenaga ahli yang ketersediaannya terbatas, terutama di daerah terpencil. Penelitian ini mengembangkan sistem klasifikasi penyakit kulit sapi menggunakan *Convolutional Neural Network* (CNN) berbasis pendekatan *transfer learning* dengan arsitektur VGG16 dan VGG19. Dataset yang digunakan terdiri dari 1.024 citra kulit sapi yang terbagi menjadi dua kelas, yaitu *Lumpy Skin* (324 citra) dan *Normal Skin* (700 citra). Pengujian dilakukan melalui enam skenario dengan variasi arsitektur dan proporsi pembagian data (70:20:10, 75:15:10, dan 80:10:10). Seluruh skenario menggunakan parameter seragam, yaitu *learning rate* 0,0001, *batch size* 32, *optimizer* Adam, dan bobot pretrained ImageNet. Hasil terbaik diperoleh pada Skenario 2A menggunakan arsitektur VGG16 dengan proporsi data 75:15:10, yang mencapai akurasi 89,22%, *macro average F1-score* 0,860, dan *precision* kelas *Lumpy Skin* sempurna sebesar 1,0000. Model terbaik selanjutnya diimplementasikan ke dalam aplikasi web interaktif menggunakan Gradio dan berhasil mengklasifikasikan 10 citra uji dengan tingkat keberhasilan 100%.

ABSTRACT

Pramitha, Arkha Yusfina. 2026. **Classification of Cattle Skin Diseases Using CNN (Convolutional Neural Network)**. Undergraduate Thesis. Department of Informatics Engineering, Faculty of Science and Technology, Maulana Malik Ibrahim State Islamic University of Malang. Supervisor: (I) Dr. Ir. Fachrul Kurniawan, M.MT., IPU (II) Shoffin Nahwa Utama, M.T

Keywords: Image Classification, Convolutional Neural Network, Lumpy Skin Disease, VGG16, VGG19, Gradio

Lumpy Skin Disease (LSD) is a contagious skin disease in cattle caused by a virus that can cause significant economic losses for farmers. Early visual detection of this disease still relies on the expertise of veterinary professionals whose availability is limited, especially in remote areas. This study develops a cattle skin disease classification system using *Convolutional Neural Network* (CNN) based on a *transfer learning* approach with VGG16 and VGG19 architectures. The dataset used consists of 1,024 cattle skin images divided into two classes, namely *Lumpy Skin* (324 images) and *Normal Skin* (700 images). Testing was carried out through six scenarios with variations in architecture and data split ratios of 70:20:10, 75:15:10, and 80:10:10. All scenarios used uniform parameters, namely a learning rate of 0.0001, batch size of 32, Adam optimizer, and ImageNet pretrained weights. The best results were obtained in Scenario 2A using the VGG16 architecture with a 75:15:10 data split ratio, achieving a accuracy of 89.22%, a macro average F1-Score of 0.860, and a perfect *Lumpy Skin* class precision of 1.000. The best model was subsequently implemented into an interactive web application using Gradio and successfully classified 10 new test images with a 100% success rate.

مستخلص البحث

فراميتا، أرخي يوسفينا، 2026. تصنيف أمراض جلد الأبقار باستخدام طريقة الشبكة العصبية الالتفافية (CNN). البحث الجامعي. قسم الهندسة المعلومات، كلية العلوم والتكنولوجيا بجامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف الأول: د. فخر الكورنياوان، الماجستير؛ المشرف الثاني: صفين نحو أوتاما، الماجستير.

الكلمات الرئيسية: تصنيف صور، شبكة عصبية الالتفافية، مرض جلد عقدي، VGG16، VGG19، Gradio.

مرض الجلد العقدي (LSD) هو مرض جلدي معدٍ يصيب الأبقار ناتج عن فيروس ويمكن أن يسبب خسائر اقتصادية كبيرة للمزارعين. يعتمد الكشف المبكر عن هذا المرض بصرياً على خبرة المتخصصين المتوفرة بشكل محدود، خاصة في المناطق النائية. هدف هذا البحث إلى تطوير نظام لتصنيف أمراض جلد الأبقار باستخدام الشبكة العصبية الالتفافية على أساس مدخل التعلم بالنقل مع بنية VGG16 و VGG19. تتكون مجموعة البيانات المستخدمة من 1,024 صورة لجلد الأبقار مقسمة إلى فئتين، وهما الجلد المتكامل (324 صورة) والجلد الطبيعي (700 صورة). وتم إجراء الاختبارات من خلال ستة سيناريوهات مع تنوعات في البنية النمطية ونسب تقسيم البيانات (75:15:10، 70:20:10، 80:10:10). جميع السيناريوهات استخدمت معايير موحدة، وهي معدل التعلم 0.0001، حجم الدفعة 32، المحسن آدم، والأوزان المدربة مسبقاً من *ImageNet*. تم الحصول على أفضل النتائج في السيناريو 3 باستخدام بنية VGG16 مع نسبة بيانات 75:15:10، حيث حقق دقة 89.22٪، ومتوسط قيمة ف1 0.860، وثبات فئة الجلد المتكامل المثالي بنسبة 1.0000. تم بعد ذلك تطبيق أفضل نموذج في تطبيق ويب تفاعلي باستخدام *Gradio*، ونجح في تصنيف 10 صور اختبار بمعدل نجاح بلغ 100٪.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sektor peternakan merupakan bagian penting pada agribisnis yang mempunyai prospek sangat menjanjikan (Sugihandono & Fatta, 2019). Kontribusinya tidak hanya berperan dalam menjaga ketahanan pangan, namun juga membantu menopang perekonomian masyarakat, terutama di daerah pedesaan (Bettencourt & Narciso, 2015). Produk peternakan menjadi salah satu sumber utama pemenuhan gizi masyarakat, karena mampu menyediakan protein hewani yang sangat dibutuhkan tubuh. Menurut data dari Kementerian Pertanian, konsumsi daging sapi di Indonesia pada tahun 2023 mencapai sekitar 2,56 kg per kapita per tahun, sementara konsumsi susu sebesar 16,27 liter per kapita per tahun (Pertanian, 2023). Angka tersebut masih tergolong rendah dibandingkan dengan negara-negara ASEAN lainnya, sehingga peningkatan produktivitas sapi sangat diperlukan untuk mengejar ketertinggalan.

Di antara berbagai jenis hewan ternak yang dipelihara, sapi menempati posisi pertama sebagai komoditas unggulan. Hal ini dapat dipahami mengingat sapi mampu memberikan berbagai manfaat ekonomi sekaligus sosial bagi peternak (Nurtini & UM, 2014). Selain menjadi sumber utama daging dan susu, sapi juga menghasilkan pupuk kandang yang dapat menunjang sektor pertanian serta berperan membantu pekerjaan di ladang, terutama pada peternakan tradisional (Kusumo et al., 2007). Badan Pusat Statistik mencatat bahwa populasi sapi potong

di Indonesia mencapai lebih dari 11,7 juta ekor, sementara sapi perah sekitar 485 ribu ekor (BPS, 2024). Jumlah ini menunjukkan bahwa sapi memiliki posisi strategis dalam mendukung ketahanan pangan nasional maupun pendapatan peternak.

Meskipun demikian, perkembangan sektor peternakan sapi masih dihadapkan pada berbagai kendala yang cukup serius. Keterbatasan pakan berkualitas, tingginya biaya produksi, serta rendahnya akses peternak terhadap teknologi modern masih menjadi kendala utama yang menghambat kemajuan sektor ini. Kekurangan pakan tidak hanya berdampak pada kesehatan ternak, tetapi juga memengaruhi aspek reproduksi dan produksi ternak, bahkan berkontribusi terhadap peningkatan emisi gas (Gloriabarus, 2023). Ketersediaan pakan berkualitas juga masih menjadi masalah serius karena dipengaruhi musim dan alih fungsi lahan (Lubis et al., 2025). Kajian menunjukkan bahwa teknologi digital seperti IoT, sensor biometrik, *big data*, dan kecerdasan buatan mampu meningkatkan efisiensi peternakan dan kesejahteraan hewan, namun adopsinya masih terkendala oleh literasi digital, infrastruktur, dan kebijakan yang belum memadai di negara berkembang (Sulistyo, 2025).

Selain kendala tersebut, berbagai penyakit ternak juga menjadi ancaman yang dapat menurunkan produktivitas dan kualitas hasil ternak. Khususnya penyakit kulit seperti *lumpy skin disease*, *scabies*, dan *ringworm* kerap menyerang sapi dan menimbulkan dampak yang cukup besar (Ambilo & Kebede, 2019). Wabah *lumpy skin disease* pertama kali muncul di Indonesia pada tahun 2022 telah menyebar ke berbagai provinsi dan menyebabkan ribuan ekor sapi terinfeksi (Sendow et al.,

2021). Kondisi ini berdampak pada penurunan bobot badan, turunnya produksi susu, hingga kematian pada kasus yang parah, sehingga menimbulkan kerugian ekonomi yang signifikan bagi para peternak.

Kondisi ini menegaskan bahwa pengembangan sistem identifikasi penyakit yang lebih efektif sangat penting untuk menjaga produktivitas sapi. Upaya peningkatan kesehatan ternak tidak hanya berdampak pada kesejahteraan peternak, tetapi juga pada ketersediaan pangan bergizi bagi masyarakat luas. Oleh karena itu, sektor peternakan sapi memerlukan perhatian khusus, baik dari sisi teknologi, kebijakan pemerintah, maupun inovasi dalam penanganan penyakit ternak.

Dalam perspektif Islam, hewan ternak merupakan anugerah sekaligus amanah dari Allah SWT yang wajib dijaga dan dirawat dengan sebaik-baiknya. Hal ini ditegaskan dalam QS. An-Nahl ayat 5:

وَالْأَنْعَامَ خَلَقَهَا ۚ لَكُمْ فِيهَا دِفْءٌ وَمَنَافِعُ وَمِنْهَا تَأْكُلُونَ

Artinya: *“Dan Dia telah menciptakan binatang ternak untuk kamu; padanya ada (bulu) yang menghangatkan dan berbagai macam manfaat, dan sebahagiannya kamu makan.”* (Q.S. An-Nahl:5)

Ayat ini menegaskan bahwa hewan ternak diciptakan Allah SWT dengan berbagai manfaat bagi manusia, sehingga menjaga kesehatannya merupakan bagian dari tanggung jawab manusia sebagai khalifah di bumi. Apabila hewan ternak mengalami gangguan kesehatan seperti penyakit kulit, maka berbagai manfaat yang seharusnya diperoleh akan terganggu baik dari sisi produksi maupun nilai ekonomisnya. Oleh karena itu, upaya mendeteksi penyakit kulit sapi secara dini melalui pengembangan teknologi merupakan bentuk nyata dari tanggung jawab tersebut. Islam juga mendorong manusia untuk tidak bersikap pasif dalam

menghadapi penyakit, melainkan aktif berikhtiar mencari solusinya, sebagaimana sabda Rasulullah SAW:

مَا أَنْزَلَ اللَّهُ دَاءً إِلَّا أَنْزَلَ لَهُ شِفَاءً

Artinya: *“Tidaklah Allah menurunkan penyakit kecuali Dia juga menurunkan penawarnya.” (HR Bukhari)*

Hadis diatas menunjukkan bahwa setiap penyakit, termasuk penyakit hewan ternak pasti memiliki solusi yang dapat ditemukan melalui usaha dan pengembangan ilmu pengetahuan. Upaya penelitian dan penerapan teknologi dalam bidang kesehatan hewan ternak merupakan bagian dari ikhtiar yang selaras dengan ajaran Islam. Pemanfaatan akal dan ilmu pengetahuan dalam mengkaji fenomena alam ini juga diperkuat dalam Q.S. Ali Imran ayat 190:

إِنَّ فِي خَلْقِ السَّمَوَاتِ وَالْأَرْضِ وَاخْتِلَافِ اللَّيْلِ وَالنَّهَارِ لَآيَاتٍ لِّأُولِي الْأَلْبَابِ ٩١

Artinya: *“Sesungguhnya dalam penciptaan langit dan bumi serta pergantian malam dan siang terdapat tanda-tanda (kebesaran Allah) bagi orang yang berakal,”*

Ayat ini memerintahkan manusia untuk mengamati dan mengkaji fenomena alam, termasuk fenomena penyakit yang tampak secara visual pada hewan ternak, sebagai bentuk penggunaan akal atas tanda-tanda kebesaran Allah SWT. Sejalan dengan nilai-nilai tersebut, pemanfaatan teknologi *Convolutional Neural Network* (CNN) dalam mengklasifikasikan penyakit kulit sapi merupakan wujud nyata ikhtiar ilmiah sekaligus bentuk tanggung jawab dalam menjaga amanah Allah SWT atas hewan ternak yang memberikan banyak manfaat bagi kehidupan manusia.

Sejalan dengan pesan Al-Qur'an dan hadits tersebut, pemanfaatan akal dan teknologi modern menjadi bagian dari ikhtiar ilmiah dalam menjawab

permasalahan di bidang peternakan. Salah satu bentuk penerapan teknologi tersebut adalah penggunaan *Convolutional Neural Network* (CNN) yang memiliki kemampuan sangat efektif dalam mendeteksi kondisi kesehatan hewan, khususnya dalam mengklasifikasi penyakit kulit sapi. CNN bekerja dengan mengekstraksi fitur-fitur penting dari citra mulai dari tekstur kulit, warna, hingga bentuk lesi yang muncul akibat infeksi. Melalui proses *convolution* dan *pooling*, CNN mampu belajar membedakan ciri-ciri visual antara kulit sapi yang sehat dan yang mengalami penyakit, sehingga sistem dapat menentukan secara otomatis apakah suatu citra termasuk kategori “normal” atau “terinfeksi”. Kemampuan ini menjadikan CNN sangat relevan untuk mendukung deteksi dini penyakit kulit sapi karena proses analisis dapat dilakukan secara cepat, konsisten, dan akurat tanpa ketergantungan penuh pada observasi manual oleh tenaga ahli.

Penerapan CNN dalam klasifikasi penyakit kulit sapi diharapkan dapat menjadi solusi inovatif dalam bidang peternakan. Akan tetapi seiring dengan kemajuan teknologi, penerapan kecerdasan buatan khususnya *Convolutional Neural Network* (CNN) telah terbukti efektif dalam pengolahan citra medis maupun veteriner. CNN mampu mengenali pola dan fitur penting dari gambar, sehingga sangat potensial digunakan untuk mengklasifikasikan penyakit kulit pada sapi. Dalam pengembangannya, terdapat berbagai arsitektur CNN yang populer, di antaranya adalah VGG16 dan VGG19. Kedua arsitektur ini memiliki prinsip yang serupa, yakni menggunakan lapisan konvolusi bertingkat untuk mengekstraksi fitur citra, tetapi berbeda pada kedalaman jumlah layer. VGG16 menggunakan 16

lapisan, sedangkan VGG19 memiliki 19 lapisan, sehingga perbedaan kompleksitas ini dapat memengaruhi akurasi, waktu pelatihan, serta kebutuhan komputasi.

Membandingkan kinerja VGG16 dan VGG19 dalam klasifikasi penyakit kulit sapi menjadi penting untuk mengetahui arsitektur mana yang lebih efisien dan tepat guna dalam konteks peternakan di Indonesia. Hasil perbandingan tersebut diharapkan dapat memberikan rekomendasi penerapan teknologi yang akurat namun tetap sesuai dengan keterbatasan infrastruktur, sehingga mampu membantu peternak dalam mendeteksi penyakit secara dini, menekan angka kerugian, dan mendukung keberlanjutan sektor peternakan sapi.

1.2 Rumusan Masalah

Bagaimana penerapan metode *Convolutional Neural Network* (CNN) dengan arsitektur VGG dalam melakukan klasifikasi penyakit kulit sapi?

1.3 Tujuan Penelitian

Penelitian ini bertujuan untuk menerapkan metode *Convolutional Neural Network* (CNN) dengan arsitektur VGG dalam melakukan klasifikasi penyakit kulit sapi sehingga diperoleh model yang mampu mengidentifikasi kondisi kulit sapi secara akurat.

1.4 Batasan Masalah

- a) Dataset merupakan citra berformat PNG berasal dari Kaggle dengan judul “*Lumpy Skin Images Dataset*”, yang telah tersedia secara publik.

- b) Penelitian ini dibatasi pada proses dua kelas klasifikasi, yaitu citra kulit sapi normal dan citra kulit sapi yang mengalami infeksi *Lumpy Skin Disease* (LSD).
- c) Kinerja model dievaluasi menggunakan *confusion matrix*, dengan indikator pengukuran berupa *accuracy*, *precision*, *recall*, dan *f1-score*.

1.5 Manfaat Penelitian

- a) Menambah khasanah ilmu pengetahuan dalam bidang teknologi pengolahan citra dan kecerdasan buatan, terutama dalam penggunaan CNN untuk klasifikasi penyakit hewan.
- b) Menjadi dasar pengembangan penelitian lebih lanjut terkait sistem deteksi penyakit ternak berbasis citra digital.
- c) Membantu peternak dan praktisi kesehatan hewan dalam mendeteksi penyakit kulit sapi secara cepat dan akurat.

BAB II

STUDI PUSTAKA

2.1 Penelitian Terkait

Beberapa penelitian terdahulu terkait dengan penggunaan convolution neural network sebagai metode menganalisis penyakit kulit pada sapi sebagai berikut.

Tabel 2.1 Penelitian Terkait

No	Judul dan author	Metode penelitian	Hasil penelitian
1	Optimisasi model <i>convolutional neural network</i> untuk klasifikasi penyakit <i>lumpy skin</i> pada sapi (Mahbub et al., 2025)	CNN dengan arsitektur DenseNet-121, MobileNetV1, MobileNetV2, GoogleNet. Dataset: 1.356 citra dari Kaggle. Evaluasi: Akurasi sebelum & sesudah segmentasi.	Hasil penelitian menunjukkan bahwa GoogleNet memiliki akurasi tertinggi yaitu 85,79% (tanpa segmentasi) dan 87,03% (dengan segmentasi). Kemudian, MobileNetV2 naik dari 82,65% menjadi 84,50%. DenseNet-121 & MobileNetV1 sedikit turun setelah segmentasi. Segmentasi tidak selalu meningkatkan akurasi.
2	Identification of Lumpy Skin Disease in Cattle with Image Classification using the Convolutional Neural Network Method (Sentoso et al., 2024)	Penelitian ini menggunakan algoritma <i>Convolutional Neural Network</i> (CNN) untuk mengklasifikasikan kesehatan sapi, khususnya mendeteksi penyakit Lumpy Skin Disease (LSD) yang penularannya sangat cepat.	Hasil penelitian menunjukkan bahwa VGG16 memberikan performa terbaik dengan akurasi pelatihan 95,31%, akurasi pengujian 96,88%, serta nilai loss yang paling rendah, sehingga dinilai paling efektif dibandingkan VGG19 dan ResNet50.
3	Detection of Lumpy Disease in Livestock Using the MobileNetV2 Architecture Method (Putra & Wiriasto, 2024)	Metode kuantitatif dengan pendekatan reuse-oriented development dan menggunakan algoritma MobileNetV2 yang dilatih dengan data augmentasi untuk klasifikasi citra penyakit LSD.	Model klasifikasi MobileNetV2 berhasil mendeteksi LSD dengan tingkat akurasi 95,91%. Aplikasi yang dikembangkan memudahkan peternak dalam mendeteksi penyakit lebih awal sehingga tindakan pencegahan dapat dilakukan lebih cepat.
4	Identifikasi Lumpy Skin Disease Menggunakan Tensorflow Dengan Metode Convolutional Neuron Network	Penelitian menggunakan <i>Convolutional Neural Network</i> (CNN) dengan arsitektur DenseNet121, yang memanfaatkan koneksi rapat antarblok sehingga setiap lapisan mendapat masukan dari semua lapisan sebelumnya. Data berupa	Model DenseNet121 berhasil mencapai akurasi 80,21% dalam mengklasifikasikan penyakit <i>Lumpy Skin Disease</i> (LSD) pada sapi, memberikan kontribusi pada efisiensi dan akurasi deteksi penyakit secara otomatis.

	(Safiq Alfiansyah & Litanianda, 2024)	citra penyakit kulit sapi dilatih menggunakan TensorFlow sebagai framework utama untuk pengembangan dan evaluasi model.	
5	Klasifikasi dan Deteksi Penyakit Kulit Melalui Pengolahan Citra dengan Metode CNN (Wona et al., 2025)	Penelitian ini menggunakan data citra penyakit kulit dari berbagai sumber. Tahapannya meliputi pengumpulan data, preprocessing untuk meningkatkan kualitas citra, dan pembuatan model CNN yang dilatih untuk klasifikasi serta deteksi penyakit kulit. Metode utama yang digunakan adalah Convolutional Neural Network (CNN).	Model CNN mampu mencapai akurasi tinggi dalam mengklasifikasikan dan mendeteksi jenis penyakit kulit. Selain itu, sistem yang dikembangkan dapat membantu diagnosis dini dan pengambilan keputusan medis secara lebih akurat.

2.2 Sapi

Sapi merupakan salah satu hewan ternak yang memiliki peran penting dalam sektor pertanian dan peternakan di Indonesia. Hewan ini menjadi sumber utama protein hewani melalui daging dan susu, serta berkontribusi pada penyediaan bahan baku industri, seperti kulit dan pupuk organik. Kesehatan sapi menjadi faktor utama dalam mendukung produktivitas dan kualitas hasil ternak (Luthfi et al., 2024).

Berbagai penyakit kulit pada sapi, seperti *dermatophilosis*, *ringworm*, atau infeksi bakteri, dapat menurunkan kualitas produksi dan mengganggu kesehatan hewan. Oleh karena itu, deteksi dini terhadap penyakit kulit sangat diperlukan agar tindakan pencegahan dan pengobatan dapat dilakukan secara cepat dan tepat. Perkembangan teknologi kecerdasan buatan, khususnya dalam pengolahan citra, memungkinkan identifikasi penyakit kulit pada sapi dilakukan secara otomatis dengan memanfaatkan data gambar.

2.3 *Lumpy Skin Disease (LSD)*

Lumpy Skin Disease merupakan penyakit kulit menular yang disebabkan oleh *Lumpy Skin Disease Virus* (LSDV) dari genus *Capripoxvirus*. Penyakit ini ditandai dengan munculnya benjolan di permukaan kulit, demam, penurunan berat badan, serta dapat menurunkan kualitas daging dan susu (Sendow et al., 2021).

Virus penyebab penyakit ini memiliki ukuran sekitar 150 kilobase pairs dengan diameter 230–260 nm (Lojkic et al., 2018). Di Indonesia, jenis sapi potong yang umum dijumpai meliputi Peranakan Ongole (PO), Bali, Madura, Brahman, Limousin, dan Simmental. Kualitas seekor sapi umumnya dinilai dari kuantitas dan kualitas karkas yang dihasilkan, dengan parameter meliputi persentase karkas, tebal lemak punggung, dan indeks perdagingan. Namun, sapi dengan bobot hidup tinggi tidak selalu memiliki persentase karkas yang tinggi, karena hal ini sangat dipengaruhi oleh bobot potong saat penyembelihan (Juandhi et al., 2019).

Sapi yang terinfeksi *Lumpy Skin Disease* (LSD) umumnya menunjukkan gejala awal berupa demam tinggi hingga 41,5°C, penurunan nafsu makan, produksi susu berkurang, ingusan, konjungtivitis, hipersalivasi, depresi, serta pembengkakan kelenjar getah bening seperti Lgl. Subscapularis dan Lgl. Prefemoral. Selain itu, muncul nodul berdiameter 2–5 cm di bawah kulit atau otot pada bagian kepala, leher, punggung, abdomen, ekor, dan organ reproduksi. Nodul ini dapat mengalami nekrosis hingga membentuk lesi dalam, yang pada sapi jantan dapat menimbulkan infertilitas permanen atau sementara, sedangkan pada sapi betina dapat menyebabkan abortus maupun infertilitas (Tawaf, 2018).

2.4 *Convolutional Neural Network (CNN)*

Convolutional Neural Network (CNN) merupakan salah satu jenis *deep learning* karena memiliki struktur jaringan yang dalam. *Deep learning* sendiri adalah cabang dari *machine learning* yang memungkinkan komputer belajar melakukan tugas layaknya manusia melalui proses *training* (Ravi & Yadhukrishna, 2020). CNN telah terbukti sangat berhasil dalam pengenalan citra dan video skala besar (Pigou et al., 2015). Selain itu, penerapan CNN telah banyak diadopsi oleh berbagai industri, seperti Amazon, Facebook, dan Google. Bahkan, Google memanfaatkan CNN untuk mengenali nomor rumah dari gambar StreetView (Lum et al., 2020). Dalam CNN, setiap neuron direpresentasikan dalam bentuk dua dimensi, sehingga metode ini sangat sesuai untuk pemrosesan data dengan input berupa citra (Cahya et al., 2021).

Berbeda dengan manusia, komputer mengenali gambar sebagai *array* dari nilai-nilai piksel. Misalnya, pada gambar dengan resolusi 260×260 piksel, akan terbentuk *array* berukuran $260 \times 260 \times 3$, di mana angka 3 mewakili tiga komponen warna RGB. *Array* ini kemudian diproses oleh CNN untuk menghasilkan probabilitas suatu gambar termasuk ke dalam kelas tertentu, seperti 96% untuk jenis Poodle, 8% untuk Spaniel, dan 0,3% untuk Pomeranian (Arnita et al., 2022).

Struktur CNN umumnya terdiri dari dua tahap utama, yaitu *feature learning* dan *classification*. Tahap *feature learning* mencakup *convolution layer*, fungsi aktivasi ReLU, dan *pooling layer*. Sementara itu, tahap *classification* meliputi proses *flatten*, *fully-connected layer*, dan prediksi. Pada setiap bagian CNN,

terdapat dua proses kunci yang berlangsung, yaitu *feed-forward* dan *backpropagation* (Yang et al., 2016).

Komponen utama CNN: *convolution layer*, *pooling layer*, *fully connected layer*, *activation function*, dan *softmax layer*.

a) *Convolution Layer*

Convolution Layer adalah komponen utama dalam CNN. Pembaruan yang dilakukan melibatkan penambahan *depth-wise convolution* di setiap saluran input dengan pola berbeda pada gambar 2D. Kombinasi kedua jenis konvolusi ini menciptakan *over-parameters*, tetapi secara matematis dapat disederhanakan menjadi satu lapisan konvolusi baru yang disebut DO-Conv. Eksperimen menunjukkan bahwa mengganti lapisan konvolusi konvensional dengan DO-Conv meningkatkan kinerja CNN dalam berbagai tugas seperti klasifikasi, deteksi, dan segmentasi. Pada tahap inferensi, *depth-wise convolution* digabungkan ke konvolusi biasa sehingga kompleksitas komputasinya tetap sama, menjadikan DO-Conv alternatif yang efisien dan bermanfaat (Ermatita et al., 2025).

b) *Pooling Layer*

Pooling Layer adalah bagian dari struktur CNN yang memproses *feature map* dengan menerapkan operasi statistik pada nilai piksel di sekitarnya. Biasanya, *pooling layer* ditempatkan setelah beberapa lapisan konvolusi untuk secara bertahap mengurangi dimensi *feature map*, jumlah parameter, dan beban komputasi, sekaligus membantu mencegah *overfitting*. Lapisan ini bekerja pada setiap *feature map* dan secara signifikan mengecilkan

dimensinya. Umumnya, digunakan filter 2x2 dengan *stride* 2, yang dapat memangkas ukuran *feature map* hingga 75% dari ukuran awal (Fhadilla, 2024).

c) *Fully Connected Layer*

Fully Connected Layer adalah lapisan yang sering digunakan pada *Multilayer Perceptron* untuk mentransformasi data menjadi bentuk yang dapat diklasifikasikan secara linear. Sebelum masuk ke lapisan ini, data dari lapisan konvolusi diubah menjadi satu dimensi. Namun, proses ini menyebabkan hilangnya sebagian informasi spesifik, sehingga Fully Connected Layer biasanya ditempatkan di bagian akhir jaringan (Peryanto et al., 2019).

2.5 Penerapan CNN Pada Klasifikasi Citra Penyakit

Convolutional Neural Network (CNN) banyak diterapkan dalam bidang medis, khususnya untuk klasifikasi citra penyakit. CNN mampu mengekstraksi fitur penting dari citra medis seperti X-ray, CT scan, MRI, atau citra mikroskopis dengan akurasi yang tinggi. Proses klasifikasi ini umumnya melibatkan beberapa tahap, yaitu:

a) Akuisisi Data

Citra medis dikumpulkan dari berbagai sumber, misalnya rumah sakit atau dataset publik. Setiap citra diberi label sesuai dengan jenis penyakitnya, misalnya pneumonia, tumor, atau kanker kulit (Sari, n.d.).

b) *Preprocessing Data*

Data citra biasanya mengalami tahap prapemrosesan seperti normalisasi ukuran, peningkatan kontras, atau augmentasi data untuk meningkatkan kualitas input ke dalam CNN.

c) Ekstraksi Fitur dengan CNN

CNN secara otomatis mengekstraksi fitur penting melalui lapisan konvolusi dan pooling. Fitur ini meliputi pola tekstur, bentuk, atau perbedaan intensitas piksel yang relevan untuk mendeteksi penyakit.

d) Klasifikasi

Setelah fitur diekstraksi, lapisan *Fully Connected* dan *Softmax* digunakan untuk mengklasifikasikan citra ke dalam kategori tertentu, misalnya sehat atau terinfeksi penyakit.

e) Evaluasi Model

Model dievaluasi menggunakan metrik seperti akurasi, sensitivitas, dan spesifisitas untuk memastikan bahwa model dapat mengidentifikasi penyakit dengan benar.

Penerapan CNN dalam klasifikasi citra penyakit telah terbukti efektif untuk mendukung diagnosis medis secara cepat dan akurat, bahkan dapat membantu tenaga medis dalam pengambilan keputusan klinis. Seperti studi dari (Khani & Rakasiwi, 2025) yang menunjukkan bahwa ResNet-50 dapat mengatasi masalah vanishing gradient dan mengekstraksi fitur kompleks dari citra medis secara optimal. Dengan kinerja tersebut, model ini dapat diaplikasikan dalam sistem kecerdasan buatan untuk membantu tenaga medis dalam mendeteksi dini penyakit

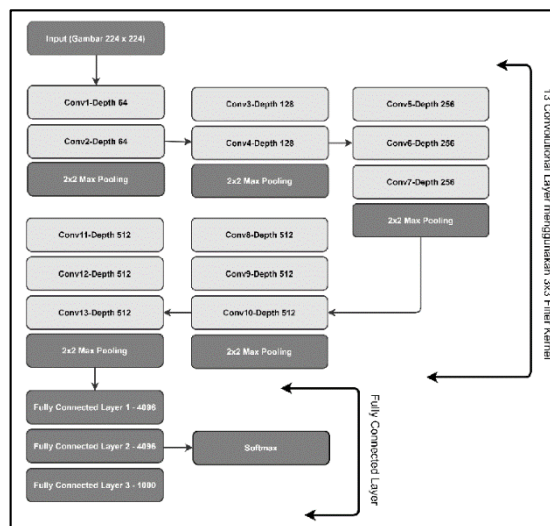
kulit secara cepat, akurat, dan efisien. Sedangkan studi lain (Shidik et al., 2024) ResNet50 dengan *learning rate* 0.01 mencapai akurasi 92,27% dalam klasifikasi penyakit mata seperti katarak, glaukoma, dan retinopati diabetik. Dibandingkan MobileNet dan DenseNet, ResNet50 lebih unggul dalam akurasi dan stabilitas, mendukung deteksi dini dan keputusan klinis untuk mencegah kebutaan.

2.6 Arsitektur VGG16 dan VGG19

Visual Geometry Group (VGG) adalah salah satu arsitektur CNN yang paling populer dan banyak digunakan. Diperkenalkan pada tahun 2014 dalam kompetisi *ImageNet Large Scale Visual Recognition Challenge*, VGG memang hanya meraih posisi runner-up, tetapi tetap mampu bersaing dengan arsitektur lain. Keunggulan VGG terletak pada kesederhanaan desainnya dan penggunaan *convolutional kernels* berukuran kecil, yang memungkinkan pembentukan jaringan lebih dalam dengan jumlah parameter relatif sedikit. Hal ini membuat proses pelatihan dan pengujian lebih efisien, serta meningkatkan kinerja pada tugas pengenalan citra. Dua varian VGG yang paling sukses adalah VGG-16 dan VGG-19.

VGG-16 merupakan salah satu arsitektur dalam *Convolutional Neural Network* (CNN) yang memiliki total 16 lapisan, terdiri atas 13 lapisan konvolusi dan 3 lapisan *fully connected* (Wardani et al., 2023). Arsitektur ini diperkenalkan oleh Karen Simonyan dan Andrew Zisserman dari Universitas Oxford pada tahun 2014 melalui makalah berjudul *Very Deep Convolutional Network for Large Scale Image Recognition*. Keunggulan utama VGG-16 terletak pada penggunaan *receptive field* konvolusi berukuran 3x3 dengan langkah 1, *padding* yang konsisten, serta *max pooling* berukuran 2x2 dengan *stride* 2, sehingga meminimalkan jumlah

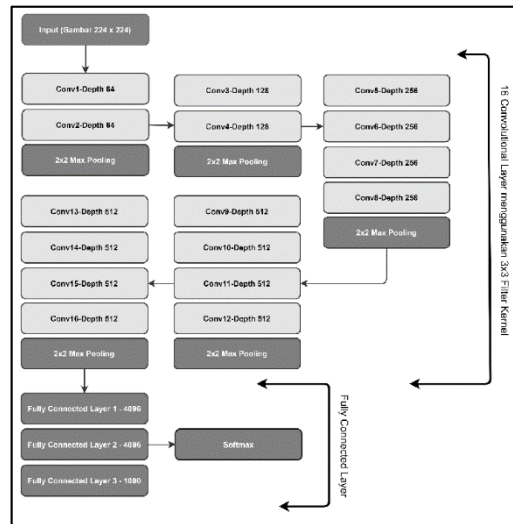
hyperparameter. Jaringan ini berukuran besar dengan sekitar 144 juta parameter, dan pada bagian akhir mencakup 3 *fully connected layer* termasuk *softmax layer* sebagai lapisan output.



Gambar 2.1 Struktur Arsitektur VGG16

Berbeda dengan VGG-16 yang memiliki 16 lapisan, VGG-19 terdiri atas 19 lapisan dalam arsitektur CNN, yaitu 16 *convolutional layer* dan 3 *fully connected layer*. Setiap dari 16 *convolutional layer* diakhiri dengan *max-pooling* yang terbagi dalam 5 grup. Sementara itu, pada *fully connected layer* terdapat dua lapisan dengan masing-masing 4096 neuron, dan satu lapisan output dengan 1000 neuron. VGG-19 mampu melatih citra dari database ImageNet yang berisi satu juta gambar dengan 1000 kategori. Arsitektur ini menggunakan filter berukuran 3x3 pada setiap convolutional layer dan merupakan pengembangan dari VGG-16 dengan total 19 weight layer untuk memperdalam jaringan. Peningkatan kedalaman ini memungkinkan VGG-19 menangkap detail gambar yang lebih kompleks, meskipun membutuhkan waktu pelatihan lebih lama. Keuntungannya adalah adanya sedikit

peningkatan akurasi berkat arsitektur yang lebih dalam (Bansal et al., 2023)(Likhar & Ridhorkar, 2024).



Gambar 2.2 Struktur Arsitektur VGG19

2.7 Evaluasi Kinerja Model

Ada beberapa cara atau metode yang dapat digunakan untuk menskalakan kinerja dari proses klasifikasi. Salah satu perhitungan yang dapat dilakukan yaitu menggunakan *confusion matrix*. *Matrix* yang paling sering digunakan di dalam proses ini yaitu *accuracy*, *precision*, *recall* dan *f-score accuracy*. Metode ini adalah metode yang paling umum digunakan untuk klasifikasi proses pada CNN arsitektur. Metode ini berisi informasi klasifikasi aktual dan prediksi oleh sistem. Contoh berikut pada Tabel 2.2 menunjukkan *matrix confusion* untuk mengklasifikasi dua kelas.

Tabel 2.2 Confusion Matrix

Gambar	Prediksi Negatif	Prediksi Positif
Negatif	TN	FP
Positif	FN	TP

Confusion Matrix sangat populer digunakan untuk memecahkan masalah klasifikasi. Kasus khusus dalam *Confusion Matrix* sering digunakan dua kelas dimana kedua kelas tersebut ditetapkan sebagai kelas positif dan kelas negatif. Kedua kelas tersebut memiliki 4 kemungkinan kombinasi yaitu *True Positif*, *True Negative*, *False Positive* dan *False Negative* seperti yang terlihat di dalam Tabel 2.2. Berdasarkan tabel tersebut, *True Positive* atau TP adalah hasil positif yang diklasifikasikan dengan benar oleh model klasifikasi. Sedangkan *True Negatif* atau TF adalah sebuah contoh gambar negatif yang benar diklasifikasikan oleh model klasifikasi. Sedangkan untuk kelas negatif yaitu *False Positive* atau FP adalah dimana gambar yang berupa kelas negatif tetapi diklasifikasikan sebagai positif. Sedangkan yang terakhir yaitu FN atau *False Negative* adalah hasil gambar yaitu negatif tetapi seharusnya adalah kelas positif.

Beberapa matriks yang sering digunakan dalam proses klasifikasi adalah *accuracy*, *precision*, *recall* dan *f1-score*.

a) *Accuracy*

Accuracy adalah proporsi jumlah total prediksi yang benar. *Accuracy* dapat tidak akurat jika digunakan dalam data yang tidak seimbang. Oleh karena itu *confusion matrix* digunakan untuk mengevaluasi kinerja dari model ini. Pada *confusion matrix*, *accuracy* mengacu pada sejauh mana prediksi dari sebuah model sesuai dengan kenyataan yang dimodelkan. Keakuratan dari suatu model dinilai dan diperkirakan dengan menerapkannya pada data pengujian yang memiliki tabel yang diketahui. Keakuratan pengklasifikasian

juga dapat dihitung dari jumlah objek yang diklasifikasikan dengan benar per total dari seluruh objek.

b) Precision

Precision menunjukkan akurat model dalam memprediksi nilai positif dan mengukur keakuratan nilai prediksi positif. Dalam matriks *precision* melibatkan *True Positive* dan *False Positif*. *Precision* yang tinggi maka lebih sedikit *false positive*.

c) Recall

Recall digunakan untuk mengukur kekuatan dari model untuk memprediksi nilai positif. *Recall* menggambarkan informasi seberapa baik model dapat menangkap semua *instance* positif. Matriks ini berkaitan dengan *True Positive* dan *False Negative*, jika nilai *recall* tinggi maka sedikit nilai *false negative*.

d) F1-Score

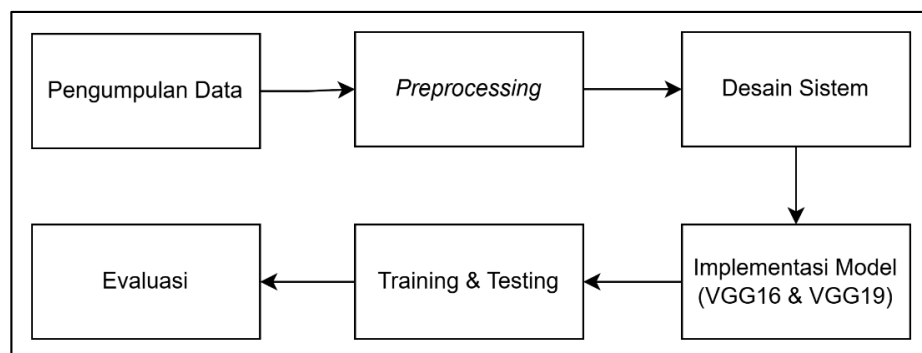
F-Measure dikenal juga sebagai *F-Value* yang menggunakan skor presisi dan *recall* hasil klasifikasi. Matriks ini mengevaluasi keakuratan prediksi dalam klasifikasi kelas. *F-Measure* didefinisikan sebagai rata-rata harmonik dari *precision* dan *recall*. Oleh karena itu nilai ini terletak di antara kedua nilai tersebut namun mendekati nilai yang lebih kecil dari kedua nilai tersebut. Ini berarti jika salah satu dari nilai *presisi* dan nilai *recall* rendah, *F1-Score* cenderung mendekati nilai minimum. Semakin tinggi nilai *F-score* semakin baik performa suatu model.

BAB III

DESAIN DAN IMPLEMENTASI

3.1 Desain Penelitian

Desain penelitian ini disusun untuk menggambarkan alur kerja penelitian secara sistematis, mulai dari tahap persiapan hingga proses evaluasi model klasifikasi penyakit kulit sapi menggunakan metode *Convolutional Neural Network* (CNN) dengan arsitektur VGG16 dan VGG19. Secara keseluruhan, alur penelitian terdiri dari enam tahapan utama, yaitu pengumpulan dataset, *preprocessing* citra, implementasi model, proses pelatihan (*training*), pengujian (*testing*), dan evaluasi model. Alur lengkap penelitian ini divisualisasikan pada Gambar 3.1.



Gambar 3.1 Tahapan Penelitian

Tahapan penelitian diawali dengan pengumpulan dataset citra kulit sapi, terdiri dari dua kelas, yaitu citra kulit sehat (normal) dan citra kulit yang terindikasi penyakit kulit (lumpy). Dataset kemudian diproses pada tahap *preprocessing*, meliputi *resize*, normalisasi, dan augmentasi citra. Proses ini bertujuan untuk menyeragamkan ukuran citra, menstandarkan nilai piksel, serta menambah variasi data latih agar model dapat belajar lebih baik dan tidak mudah mengalami *overfitting*.

Setelah *preprocessing* selesai, dataset dibagi menjadi tiga bagian, yaitu data latih (*training set*), data validasi (*validation set*), dan data uji (*testing set*). Selanjutnya dilakukan perancangan sistem yang menggambarkan alur kerja mulai dari data masuk, proses *preprocessing*, pemrosesan menggunakan model CNN, hingga menghasilkan keluaran berupa prediksi kelas. Kedua arsitektur CNN, yaitu VGG16 dan VGG19, kemudian diimplementasikan dan dilatih secara terpisah menggunakan data latih, sementara data validasi digunakan untuk memantau performa model pada setiap *epoch* guna mencegah *overfitting*. Tahap akhir adalah pengujian menggunakan data uji dan evaluasi performa model menggunakan metrik akurasi, *precision*, *recall*, *F1-score*, dan *confusion matrix*, sehingga dapat ditentukan arsitektur yang paling optimal.

3.2 Pengumpulan Data

Pada penelitian ini digunakan dataset citra penyakit kulit sapi yang berasal dari “*Lumpy Skin Images Dataset*” yang diperoleh dari situs website Kaggle. Dataset tersebut terdiri dari 1.024 citra yang terbagi menjadi dua kelas, yaitu kelas normal (kulit sapi sehat) dan kelas lumpy (kulit sapi terindikasi penyakit kulit). Seluruh citra memiliki format .PNG dengan ukuran 256×256 piksel. Contoh beberapa citra pada masing-masing kelas dapat dilihat pada Tabel 3.1.

Tabel 3.1 Beberapa Contoh Dataset Penyakit Kulit Sapi

No	Kelas Citra	Citra	
1	Normal		
2	Normal		
3	Normal		
4	Lumpy		
5	Lumpy		
6	Lumpy		

Dataset ini terdiri dari dua kategori citra yang dijelaskan sebagai berikut:

- a) Kelas Normal, berjumlah 700 citra kulit sapi sehat yang tidak menunjukkan gejala penyakit kulit.

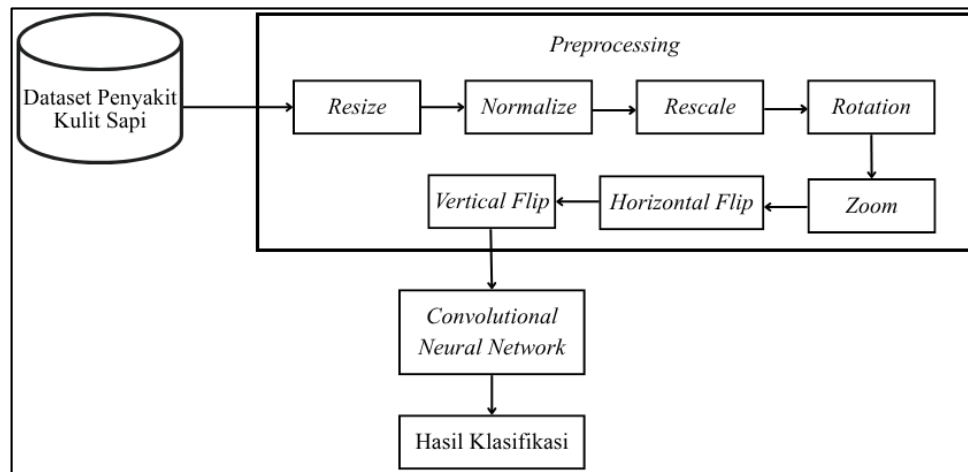
- b) Kelas Lumpy, berjumlah 324 citra kulit sapi yang menampilkan gejala penyakit kulit seperti nodul, lesi, atau perubahan tekstur kulit.

Terdapat ketidakseimbangan jumlah data antara kedua kelas, di mana kelas normal memiliki jumlah citra yang jauh lebih banyak dibandingkan kelas lumpy. Kondisi ketidakseimbangan ini berpotensi menyebabkan model cenderung lebih bias dalam memprediksi kelas mayoritas. Oleh karena itu, ketidakseimbangan tersebut diatasi melalui teknik augmentasi pada tahap *preprocessing*, sehingga variasi data pada kelas minoritas dapat ditingkatkan dan model dapat belajar secara lebih seimbang dari kedua kelas.

Seluruh citra dalam dataset dibagi menjadi tiga bagian, yaitu data latih (*training set*), data validasi (*validation set*), dan data uji (*testing set*). Data latih digunakan untuk melatih bobot model, sedangkan data uji digunakan untuk mengevaluasi performa akhir model secara objektif. Pembagian bertujuan untuk memastikan bahwa proses pelatihan dan pengujian model berlangsung secara terpisah dan independen.

3.3 Desain Sistem

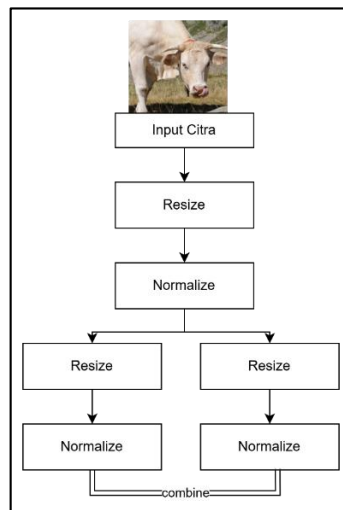
Struktur desain sistem pada penelitian ini menggambarkan alur proses yang akan dilakukan oleh sistem dalam mengklasifikasikan penyakit kulit sapi secara menyeluruh, mulai dari tahap input hingga menghasilkan output. Desain sistem pada penelitian ini disajikan pada Gambar 3.2.



Gambar 3.2 Desain Sistem

Gambar 3.2 menunjukkan alur proses sistem klasifikasi citra penyakit kulit sapi menggunakan metode *Convolutional Neural Network* (CNN). Sistem ini dirancang untuk mengidentintifikasi kondisi kulit sapi melalui citra, baik dalam keadaan normaul maupun terindikasi adanya penyakit kulit. Secara umum, tahapan sistem terdiri atas pengumpulan dataset, *preprocessing* citra, implementasi klasifikasi menggunakan CNN, serta hasil klasifikasi.

Alur proses pengolahan data secara lebih rinci ditampilkan pada Gambar 3.3. Proses diawali dari input citra yang kemudian dibagi menjadi tiga bagian, yaitu data latih, data validasi, dan data uji. Seluruh citra kemudian memasuki tahap *preprocessing* untuk dilakukan pengolahan awal. Khusus untuk data latih, diterapkan proses *resize*, normalisasi, dan augmentasi yang meliputi *rotate*, *zoom*, *vertical flip*, dan *horizontal flip*. Sementara itu, data uji hanya diterapkan proses *resize* dan normalisasi tanpa augmentasi, agar evaluasi model mencerminkan kondisi data yang sesungguhnya. Setelah *preprocessing* selesai, data diproses menggunakan dua arsitektur CNN, yaitu VGG16 dan VGG19, untuk menghasilkan prediksi kelas, yaitu normal atau lumpy.



Gambar 3.3 Proses Pengolahan Data

3.4 *Image Preprocessing*

Preprocessing merupakan tahap pengolahan awal citra yang bertujuan untuk mempersiapkan data agar siap digunakan dalam proses pelatihan dan pengujian model. Tahap ini terdiri dari dua jenis proses, yaitu proses standarisasi yang diterapkan pada seluruh data, serta augmentasi yang hanya diterapkan pada data latih untuk memperkaya variasi data tanpa mengubah karakteristik data validasi dan data uji.

3.4.1 **Resize**

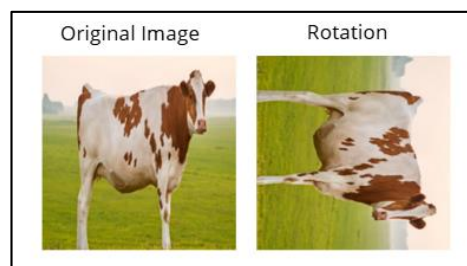
Resize bertujuan untuk menyeragamkan ukuran citra pada dataset, agar sesuai dengan kebutuhan masukan model yang akan digunakan, yaitu VGG16 dan VGG19. Kedua arsitektur tersebut membutuhkan ukuran masukan citra sebesar 224 x 224 piksel dengan tiga kanal warna (RGB). Oleh karena itu, seluruh citra pada dataset diubah ukurannya menjadi 224 x 224 agar model dapat memproses data secara konsisten.

3.4.2 Normalisasi

Setelah proses *resize*, citra dinormalisasi dengan cara mengubah nilai piksel dari rentang 0-255 menjadi 0-1. Normalisasi dilakukan untuk menstabilkan proses pelatihan, mempercepat konvergen, dan membantu model belajar lebih efisien karena nilai input yang lebih kecil dan seragam cenderung menghasilkan gradien yang lebih stabil selama proses *backpropagation*.

3.4.3 Rotate

Rotate merupakan teknik yang digunakan untuk memutar citra secara acak sesuai rentang derajat rotasi yang telah ditentukan. Teknik ini bertujuan agar model dapat mengenali pola penyakit kulit sapi dari berbagai sudut pandang, sehingga kemampuan generalisasi model terhadap variasi orientasi pengambilan gambar meningkat. Gambar 3.4 merupakan contoh hasil *rotate*.



Gambar 3.4 *Rotate*

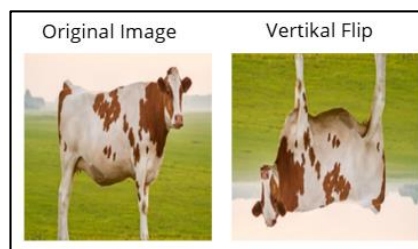
3.4.4 Zoom

Zoom digunakan untuk memperbesar citra secara acak berdasarkan nilai skala yang telah ditentukan. Teknik ini membantu model untuk mengenali fitur-fitur penyakit kulit sapi pada berbagai skala dan jarak pengambilan gambar, sehingga model lebih robust terhadap variasi jarak kamera dengan objek. Contoh hasil teknik *zoom* ditampilkan pada Gambar 3.5.

Gambar 3.5 *Zoom*

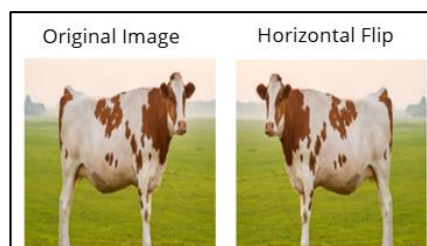
3.4.5 *Vertical Flip*

Vertical flip berfungsi untuk membalik citra secara vertikal, yaitu mencerminkan citra terhadap sumbu horizontal. Teknik augmentasi ini menghasilkan perspektif baru dari citra yang sama sehingga menambah variasi data latih tanpa memerlukan pengambilan gambar tambahan. Contoh hasilnya dapat dilihat pada Gambar 3.6.

Gambar 3.6 *Vertical Flip*

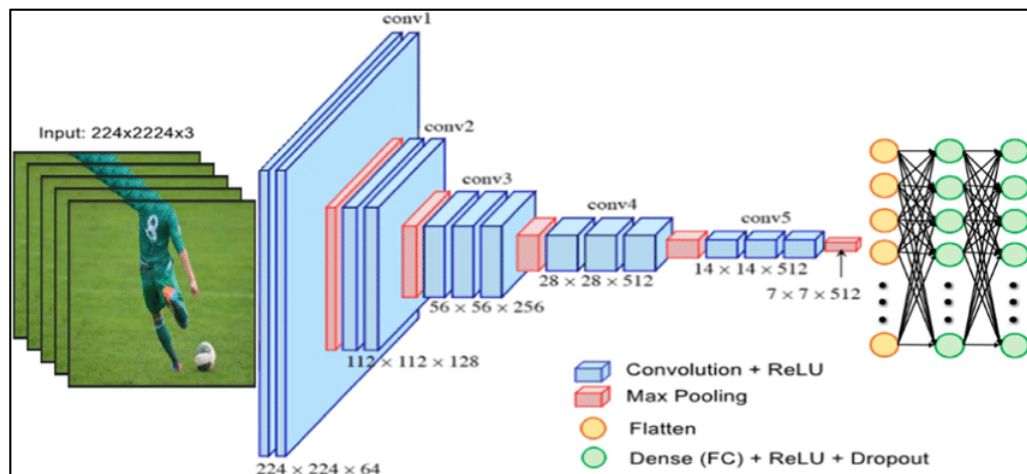
3.4.6 *Horizontal Flip*

Horizontal flip berfungsi untuk membalik citra secara horizontal, yaitu mencerminkan citra terhadap sumbu vertical. Contoh hasilnya dapat dilihat pada Gambar 3.7.

Gambar 3.7 *Horizontal Flip*

3.5 Implementasi Metode

Pada tahapan implementasi metode, data yang telah melalui proses *preprocessing* akan diproses menggunakan metode CNN. Pada penelitian ini, arsitektur CNN yang digunakan adalah VGG16 dan VGG19, yang dipilih karena memiliki kedalaman jaringan yang mampu mengekstraksi fitur visual secara efektif. Kedua arsitektur ini diuji secara terpisah untuk mengetahui performanya dalam mengklasifikasikan penyakit kulit sapi ke dalam dua kelas, yaitu normal dan lumpy.



Gambar 3.8 Arsitektur CNN VGG16 dan VGG19 (Sumber: Bilal, 2022)

Struktur arsitektur CNN yang digunakan ditunjukkan pada Gambar 3.8. Secara garis besar, arsitektur VGG16 dan VGG19 dibagi menjadi dua bagian utama, yaitu *feature extraction layer* dan *classification layer*. *Feature extraction layer* terdiri dari rangkaian *convolutional layer* dan *max pooling layer* yang bertugas mengekstraksi fitur-fitur penting dari citra secara bertahap. *Classification layer* terdiri dari beberapa *fully connected layer* yang berfungsi mengolah fitur hasil ekstraksi menjadi keputusan klasifikasi akhir. Perbedaan utama antara VGG16 dan

VGG19 terletak pada jumlah *convolutional layer*, di mana VGG16 memiliki 13 lapisan konvolusi sedangkan VGG19 memiliki 16 lapisan konvolusi.

3.5.1 Input Layer

Lapisan pertama berfungsi sebagai input layer yang menerima citra hasil *preprocessing*. Citra yang digunakan berukuran 224×224 piksel dengan tiga kanal warna (RGB), sehingga setiap citra direpresentasikan sebagai array tiga dimensi berukuran $(224 \times 224 \times 3)$. Representasi numerik ini selanjutnya dilanjutkan ke lapisan berikutnya untuk diproses lebih lanjut.

3.5.2 Convolution Layer

Convolution layer merupakan lapisan inti pada CNN untuk mengekstraksi fitur visual dari citra. Pada lapisan ini dilakukan operasi konvolusi menggunakan filter (kernel) berukuran 3×3 . Setiap *kernel* akan bergerak di seluruh area citra untuk menghasilkan *feature maps* yang merepresentasikan pola tertentu, seperti tepi, tekstur, atau bentuk. Operasi konvolusi dapat dihitung melalui Persamaan 3.1.

$$c(a, b) = \sum_{i=0}^{k_h-1} \sum_{j=0}^{k_w-1} X(a+i, b+j) W(i, j) \quad (3.1)$$

Keterangan:

$c(a, b)$ = hasil konvolusi pada posisi (a,b)

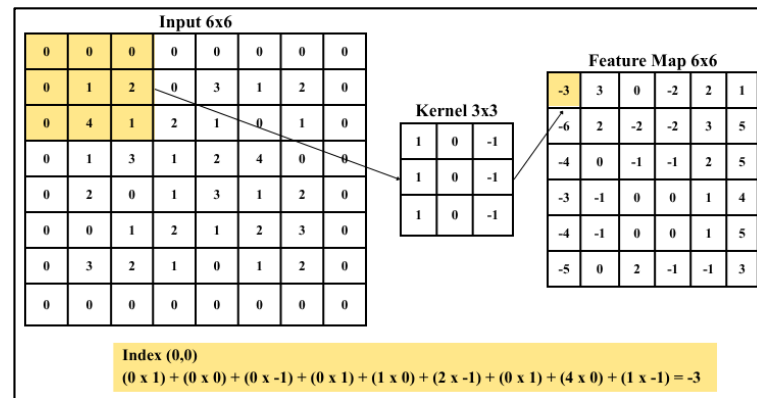
$X(a, b)$ = nilai piksel pada posisi (a,b)

$W(i, j)$ = nilai kernel pada indeks (i,j)

k_h, k_w = tinggi dan lebar kernel

Baik VGG16 maupun VGG19 menggunakan kernel 3×3 dengan *padding* “same”, sehingga ukuran output tetap terjaga. Jumlah convolution layer pada VGG16 sebanyak 13 lapisan, sedangkan VGG19 memiliki 16 lapisan convolutional

sehingga lebih dalam. Berikut contoh perhitungan manual proses convolutional pada input gambar berukuran 6x6 dengan padding same dan kernel berukuran 3x3, disajikan pada Gambar 3.9.



Gambar 3.9 Perhitungan Proses Konvolusi

Setelah proses konvolusi selesai dilakukan, setiap nilai pada *feature map* diproses menggunakan fungsi aktivasi ReLU (*Rectified Linear Unit*). Fungsi ini bekerja dengan mengubah nilai negative menjadi nol, sementara nilai positif tetap dipertahankan. Fungsi aktivasi ReLU dinyatakan pada Persamaan 3.2.

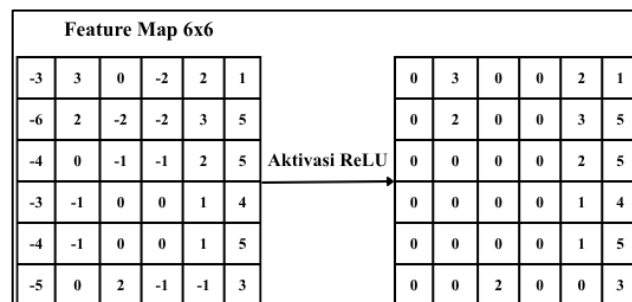
$$f(x) = \max(0, x) \quad (3.2)$$

Keterangan:

$f(x)$: hasil dari fungsi aktivasi ReLU

x : nilai hasil konvolusi pada feature map

Contoh penerapan fungsi aktivasi ReLU dapat dilihat pada Gambar 3.10.



Gambar 3.10 Penerapan Fungsi Aktivasi ReLU

3.5.3 Max Pooling Layer

Setelah melewati *convolution layer*, *feature map* yang dihasilkan akan diproses pada *max pooling layer*. Lapisan ini bertujuan untuk mengurangi dimensi *feature maps* sambil mempertahankan fitur paling penting. Operasi *pooling* dilakukan menggunakan jendela berukuran 2×2 dengan *stride* 2, di mana setiap pergeseran mengambil nilai maksimum dari area 2×2 piksel. Perhitungan *max pooling* dapat dilihat pada Persamaan 3.3.

$$m(p, q) = \max(c(ax + p, by + q)) \quad (3.3)$$

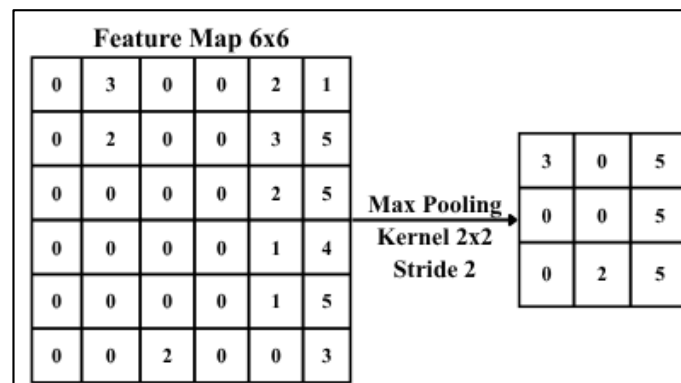
Keterangan:

$m(p, q)$ = nilai *output pooling*

$c(a, b)$ = nilai *feature maps* hasil konvolusi

p, q = indeks iterasi pada jendela *pooling*

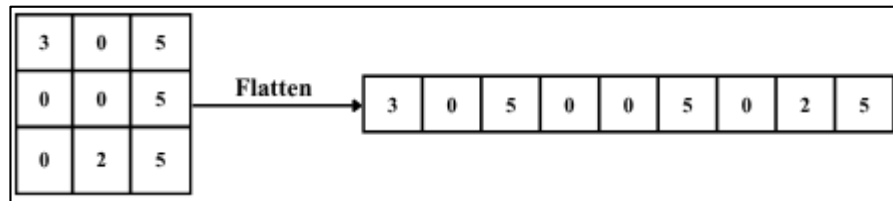
Contoh penerapan *max pooling* dapat dilihat pada Gambar 3.11.



Gambar 3.11 Penerapan *Max Pooling*

Proses *pooling* membantu mengurangi beban komputasi, memperkecil ukuran representasi fitur, sekaligus meningkatkan ketahanan model terhadap pergeseran kecil pada posisi fitur dalam citra. Pada VGG16 dan VGG19, proses ini dilakukan setelah beberapa *convolution layer*. Setelah seluruh lapisan konvolusi dan *pooling* dilewati, output akan melalui proses *flattening* menjadi vektor satu dimensi untuk

diproses oleh *fully connected layer*. Berikut hasil dari proses flattening dapat dilihat pada Gambar 3.12.



Gambar 3.12 Penerapan *Flattening*

3.5.4 *Fully Connected Layer*

Fully connected layer berfungsi untuk mengubah fitur hasil *flatten* menjadi keputusan klasifikasi. Pada bagian ini, dilakukan proses penghitungan aktivasi melalui jaringan neuron yang saling terhubung. Lapisan *fully connected* pada arsitektur VGG terdiri dari tiga lapisan, yaitu:

- a) *Fully Connected Layer* Pertama, menerima vektor satu dimensi hasil *flattening* yang kemudian diproses menggunakan fungsi aktivasi ReLU. Lapisan ini berfungsi sebagai lapisan tersembunyi pertama yang mempelajari representasi fitur tingkat tinggi dari data.
- b) *Fully Connected Layer* Kedua, memproses keluaran dari lapisan sebelumnya kembali menggunakan fungsi aktivasi ReLU untuk memperdalam pemahaman model terhadap pola yang membedakan kedua kelas.
- c) *Output Layer*, merupakan lapisan terakhir yang terdiri dari dua neuron sesuai dengan jumlah kelas pada penelitian ini, yaitu kelas normal dan lumpy. Lapisan ini menggunakan fungsi aktivasi *softmax* yang menghasilkan distribusi probabilitas untuk setiap kelas, di mana jumlah

seluruh nilai probabilitas selalu sama dengan satu. Perhitungan softmax dapat dilihat pada Persamaan 3.4.

$$f(p_i) = \frac{\exp(p_i)}{\sum_{j=1}^2 \exp(p_j)} \quad (3.4)$$

$$\hat{y} = f(p_i)$$

Keterangan:

f = fungsi aktivasi softmax

p = input

$\hat{y}$ = hasil prediksi

Fungsi softmax menghasilkan nilai probabilitas pada masing-masing kelas, dan kelas dengan probabilitas tertinggi akan dipilih sebagai hasil prediksi.

3.5.5 Loss Function

Untuk mengukur tingkat kesalahan prediksi model selama proses pelatihan, digunakan fungsi *loss* berupa *Categorical Cross-Entropy*. Pemilihan fungsi *loss* ini didasarkan pada dua pertimbangan utama. Pertama, *output layer* menggunakan fungsi aktivasi *softmax* dengan dua neuron keluaran. Kedua, label data direpresentasikan dalam format *one-hot encoding*, di mana kelas normal direpresentasikan sebagai [1, 0] dan kelas lumpy sebagai [0, 1]. *Categorical Cross-Entropy* mengukur seberapa jauh distribusi probabilitas prediksi model dari distribusi label sebenarnya. Semakin kecil nilai *loss* yang dihasilkan, semakin baik kemampuan model dalam melakukan prediksi. Rumus *Categorical Cross-Entropy* dinyatakan pada Persamaan 3.5.

$$E = - \sum_{j=0}^2 y_j \log(\hat{y}_j) \quad (3.5)$$

Keterangan:

y = nilai target (ground truth)

$\hat{y}$ = nilai prediksi model

Nilai *loss* yang dihasilkan pada setiap iterasi pelatihan digunakan dalam proses *backpropagation* untuk memperbarui bobot jaringan secara bertahap hingga model mencapai kondisi optimal.

3.6 Proses Training

Proses training merupakan tahapan penting dalam penelitian ini yang bertujuan untuk melatih model *Convolutional Neural Network* (CNN) menggunakan data training yang telah melalui proses *preprocessing*. Pada penelitian ini digunakan dua arsitektur CNN, yaitu VGG16 dan VGG19. Kedua arsitektur ini dilatih secara terpisah untuk mengetahui performanya dalam mengklasifikasikan penyakit kulit sapi ke dalam dua kategori, yaitu normal dan lumpy. Agar proses training dapat berjalan dengan baik, dibutuhkan beberapa parameter pelatihan seperti *epoch*, *batch size*, dan nilai *learning rate* yang digunakan oleh *optimizer*. Parameter-parameter ini bersifat konstan selama proses pelatihan berlangsung, sedangkan nilai bobot dan bias pada model akan diperbarui secara bertahap melalui mekanisme *backpropagation*.

Selama proses training berlangsung, citra hasil *preprocessing* dimasukkan ke dalam model melalui lapisan input. Kemudian data tersebut diproses secara bertahap melalui *convolution layer*, *activation layer*, *max pooling layer*, *flatten layer*, hingga akhirnya memasuki *fully connected layer*. Pada bagian output, model

akan menghasilkan nilai prediksi untuk kedua kelas menggunakan fungsi aktivasi softmax. Nilai prediksi ini kemudian dibandingkan dengan label sebenarnya menggunakan fungsi loss berupa *Binary Cross-Entropy*. Fungsi loss ini menghitung selisih antara nilai prediksi dan nilai target, sehingga diperoleh nilai error yang menggambarkan seberapa jauh prediksi model dari nilai yang seharusnya.

Nilai *error* tersebut akan digunakan pada tahap *backpropagation* untuk memperbarui bobot dan bias. Proses *backpropagation* bekerja dengan cara menghitung gradien dari fungsi *loss* terhadap bobot pada setiap lapisan. Gradien ini kemudian digunakan untuk memperbarui bobot menggunakan Persamaan 3.6.

$$W_{new} = W - \eta \frac{dL}{dW} \quad \text{dan} \quad b_{new} = b - \eta \frac{dL}{db} \quad (3.6)$$

Dimana η merupakan nilai learning rate, $\frac{dL}{dW}$ adalah turunan loss terhadap bobot, dan $\frac{dL}{db}$ adalah turunan loss terhadap bias. Semakin kecil nilai error yang dihasilkan, semakin kecil pula perubahan bobot yang dilakukan, sehingga model secara bertahap akan mendekati kondisi optimal. Selain itu, pada proses *feedforward*, nilai output dihitung menggunakan fungsi softmax seperti pada Persamaan 3.4.

Hasil dari fungsi ini berupa *probabilitas* pada masing-masing kelas, sehingga model dapat menentukan kelas dengan *probabilitas* tertinggi sebagai hasil prediksi. Setiap *epoch* pelatihan terdiri dari dua proses utama, yaitu *feedforward* dan *backpropagation*, yang dilakukan berulang hingga model mencapai akurasi terbaiknya. Pada penelitian ini digunakan optimizer Adam karena kemampuannya

yang adaptif dalam menyesuaikan nilai *learning rate* selama proses pelatihan, sehingga proses konvergensi dapat tercapai dengan lebih cepat dan stabil.

Selama pelatihan, nilai akurasi dan loss pada data *training* dan *validation* akan dicatat pada setiap *epoch* untuk mengamati perkembangan performa model. Nilai-nilai ini juga digunakan untuk memantau potensi *overfitting* maupun *underfitting* yang mungkin terjadi selama proses pelatihan berlangsung. Setelah model selesai dilatih, bobot terbaik akan disimpan dan digunakan pada tahap pengujian. Model yang menghasilkan performa terbaik akan dijadikan acuan untuk proses evaluasi selanjutnya.

3.7 Skenario Uji Coba

Untuk mengevaluasi performa model, penelitian ini menerapkan dua skenario uji coba yang didasarkan pada perbedaan pembagian data dan variasi arsitektur *Convolutional Neural Network* (CNN). Pada setiap skenario, model akan melalui proses pelatihan dan pengujian untuk menilai pengaruh parameter yang digunakan terhadap kinerjanya. Uji coba ini bertujuan untuk mengidentifikasi pengaruh pembagian data dan perubahan arsitektur model dalam menentukan konfigurasi terbaik, sehingga diperoleh hasil klasifikasi penyakit kulit sapi yang optimal.

Skenario pertama difokuskan pada analisis pengaruh pembagian dataset terhadap performa model. Tiga variasi proporsi yang digunakan adalah 70:20:10, 75:15:10, dan 80:10:10, yang masing-masing merepresentasikan rasio data latih, data validasi, dan data uji. Skenario variasi dataset disajikan dalam Tabel 3.2.

Tabel 3.2 Skenario Pengujian Dengan Variasi Dataset

Skenario	Data Latih	Data Validasi	Data Uji
1	70%	20%	10%
2	75%	15%	10%
3	80%	10%	10%

Skenario kedua berfokus pada pengaruh perubahan arsitektur *Convolutional Neural Network* (CNN) terhadap performa klasifikasi. Pada skenario ini, uji coba dilakukan dengan membandingkan dua arsitektur model, yaitu VGG16 dan VGG19, yang memiliki perbedaan pada jumlah lapisan. Proses pelatihan dan pengujian diterapkan pada setiap model untuk menganalisis bagaimana variasi jumlah lapisan konvolusi pada VGG16 dan VGG19 memengaruhi kemampuan model dalam ekstraksi fitur dan klasifikasi penyakit kulit sapi. Skenario arsitektur disajikan dalam Tabel 3.3.

Tabel 3.3 Skenario Pengujian Dengan Variasi Arsitektur

Skenario	Arsitektur
A	VGG16
B	VGG19

Berdasarkan kombinasi skenario yang telah disusun, dihasilkan beberapa variasi uji coba model yang bertujuan untuk mengidentifikasi kombinasi dengan kinerja model paling optimal. Kombinasi skenario uji coba disajikan dalam Tabel 3.4.

Tabel 3.4 Kombinasi Skenario Uji Coba

Kombinasi	Kombinasi Skenario Uji Coba
1A	Pembagian dataset 70:20:10 dengan arsitektur VGG16
1B	Pembagian dataset 70:20:10 dengan arsitektur VGG19
2A	Pembagian dataset 75:15:10 dengan arsitektur VGG16
2B	Pembagian dataset 75:15:10 dengan arsitektur VGG19
3A	Pembagian dataset 80:10:10 dengan arsitektur VGG16
3B	Pembagian dataset 80:10:10 dengan arsitektur VGG19

Rangkaian skenario uji coba yang dirancang diharapkan dapat memberikan gambaran yang sistematis mengenai pengaruh variasi pembagian dataset dan arsitektur CNN terhadap model klasifikasi penyakit kulit sapi. Hasil dari setiap kombinasi pengujian akan dianalisis dan dibandingkan untuk menentukan konfigurasi yang paling efektif dan stabil dalam menghasilkan performa klasifikasi terbaik, yang kemudian digunakan sebagai dasar pada tahap evaluasi dan pembahasan hasil penelitian.

3.8 Evaluasi Model

Kinerja suatu model dapat dievaluasi menggunakan beberapa metric, yaitu *accuracy*, *precision*, *recall*, dan *f1-score*, yang dihitung berdasarkan hasil *confusion matrix*. *Confusion matrix* merepresentasikan empat komponen utama, yaitu *True Positive* (TP) ketika model berhasil memprediksi kelas positif secara benar, *True Negative* (TN) ketika prediksi kelas negative sesuai dengan kondisi sebenarnya, *False Positive* (FP) ketika model memprediksi kelas positif namun kondisi sebenarnya negative, serta *False Negative* (FN) ketika model memprediksi kelas negative padahal kondisi sebenarnya positif. Berdasarkan nilai-nilai tersebut, metric *accuracy* merepresentasikan tingkat keakuratan model dalam proses klasifikasi, dihitung menggunakan Persamaan 3.7.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.7)$$

Precision digunakan untuk mengukur tingkat kesesuaian model dalam memprediksi kelas positif, yaitu perbandingan antara jumlah prediksi positif yang

benar dengan seluruh prediksi positif yang dihasilkan oleh model. *Precision* dihitung menggunakan Persamaan 3.8.

$$Precision = \frac{TP}{TP + FP} \quad (3.8)$$

Recall digunakan untuk mengukur kemampuan model dalam mendeteksi seluruh kejadian positif yang ada, yaitu dengan membandingkan jumlah prediksi positif yang benar terhadap total kejadian positif yang sebenarnya. *Recall* dihitung menggunakan Persamaan 3.9.

$$Recall = \frac{TP}{TP + FN} \quad (3.9)$$

F1-score digunakan untuk menghitung keseimbangan antara nilai *precision* dan *recall* dalam mengevaluasi kinerja model klasifikasi. *F1-score* dihitung menggunakan Persamaan 3.10.

$$F1 - score = 2 \times \frac{precision \times recall}{precision + recall} \quad (3.10)$$

BAB IV

HASIL DAN PEMBAHASAN

Bab ini menjelaskan hasil implementasi dan pengujian sistem klasifikasi penyakit kulit sapi *Lumpy Skin Disease* (LSD) menggunakan metode *Convolutional Neural Network* (CNN). Pengujian dilakukan melalui enam skenario percobaan dengan kombinasi dua arsitektur, yaitu VGG16 dan VGG19, serta variasi proporsi pembagian data yang berbeda. Hasil setiap skenario dievaluasi menggunakan *Confusion Matrix* dan diakhiri dengan pengujian model melalui antarmuka berbasis web secara *real-time*.

4.1 Uji Coba

Uji coba dilakukan untuk menilai performa model CNN dalam mengklasifikasikan penyakit kulit sapi berdasarkan gambar. Uji coba bertujuan untuk mengetahui kombinasi arsitektur dan proporsi pembagian data yang menghasilkan performa terbaik.

Dataset terdiri dari 1.024 citra yang terbagi dalam dua kelas, yaitu kelas *normal* sebanyak 700 gambar dan kelas *lumpy* sebanyak 324 gambar. Seluruh citra dibagi sesuai dengan proporsi pembagian pada skenario yang telah ditentukan.

Seluruh skenario dilatih menggunakan konfigurasi parameter yang sama. Parameter yang diterapkan meliputi *batch size*, yaitu jumlah data yang diproses sekaligus dalam satu iterasi pelatihan model; *dropout*, yaitu teknik menonaktifkan neuron secara acak guna mencegah overfitting; *learning rate*, yaitu besaran pembaruan bobot selama proses pelatihan berlangsung; serta *epoch*, yaitu jumlah

iterasi pelatihan terhadap keseluruhan dataset. Rincian parameter pelatihan disajikan pada tabel 4.1 berikut.

Tabel 4.1 Nilai Parameter

Parameter	Nilai
<i>Batch Size</i>	32
<i>Drop Out</i>	0.5
<i>Learning Rate</i>	0.0001
<i>Epoch</i>	25

Penelitian ini mencakup enam skenario percobaan, yaitu skenario 1A, 1B, 2A, 2B, 3A, dan 3B. Setiap skenario dibedakan berdasarkan dua aspek utama, yakni komposisi pembagian dataset pelatihan dan uji serta arsitektur CNN yang diterapkan. Detail konfigurasi masing-masing skenario disajikan pada tabel 4.2 berikut.

Tabel 4.2 Skenario Uji Coba

Kombinasi	Kombinasi Skenario Uji Coba
1A	Pembagian dataset 70:20:10 dengan arsitektur VGG16
1B	Pembagian dataset 70:20:10 dengan arsitektur VGG19
2A	Pembagian dataset 75:15:10 dengan arsitektur VGG16
2B	Pembagian dataset 75:15:10 dengan arsitektur VGG19
3A	Pembagian dataset 80:10:10 dengan arsitektur VGG16
3B	Pembagian dataset 80:10:10 dengan arsitektur VGG19

4.2 Hasil Uji Coba

Bagian ini memaparkan hasil pelatihan dan evaluasi model dari keenam skenario yang telah diujicobakan. Setiap skenario dievaluasi berdasarkan nilai *validation accuracy* dan *validation loss* yang diperoleh selama proses pelatihan, serta confusion matrix yang digunakan untuk mengukur kemampuan model dalam mengklasifikasikan setiap kelas secara lebih rinci. Hasil tersebut kemudian dibandingkan antar skenario untuk menentukan kombinasi arsitektur dan proporsi pembagian data yang paling optimal dalam mengklasifikasikan penyakit kulit sapi.

4.2.1 Uji Coba Skenario 1A

Skenario 1A merupakan skenario pertama dalam pengujian model klasifikasi *Lumpy Skin Disease* (LSD) menggunakan arsitektur VGG16 dengan rasio pembagian data 70:20:10, yaitu 70% data latih, 20% data validasi, dan 10% data uji.

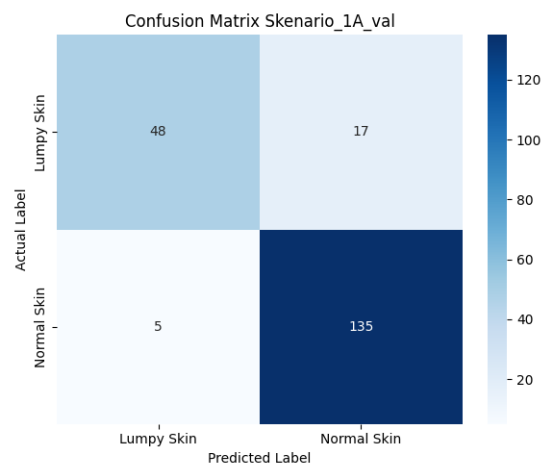


Gambar 4.1 Kurva *Accuracy* dan *Loss* Skenario 1A

Berdasarkan grafik *training history* pada Gambar 4.1, proses pelatihan Skenario 1A berlangsung selama 25 epoch. Pada epoch pertama, akurasi *training* dimulai pada nilai 71,83%, kemudian meningkat secara bertahap hingga mencapai 92,47% pada epoch ke-25. Nilai akurasi validasi (*val_accuracy*) menunjukkan pola yang lebih stabil sejak awal pelatihan, dengan nilai awal 87,80% dan nilai tertinggi 91,71% yang dicapai pada epoch ke-14, sementara nilai akhir pada epoch ke-25 berada di angka 89,27%. Pada grafik *loss*, nilai *training loss* menunjukkan tren penurunan yang konsisten dari 0,6400 pada epoch pertama hingga 0,1934 pada epoch terakhir. Nilai *validation loss* juga cenderung menurun dari 0,3758 menjadi 0,3471, meskipun sempat mengalami fluktuasi pada beberapa epoch tengah. Terdapat gap yang semakin melebar antara *training loss* dan *validation loss* menjelang akhir pelatihan, yang mengindikasikan adanya kecenderungan

overfitting ringan. Namun demikian, model masih menunjukkan generalisasi yang cukup baik mengingat nilai akurasi validasi tetap berada di atas 89% pada epoch terakhir.

Evaluasi terhadap data validasi menghasilkan *confusion matrix* sebagaimana ditampilkan pada Gambar 4.2. Dari total 205 sampel validasi yang terdiri dari 65 sampel *Lumpy Skin* dan 140 sampel *Normal Skin*, model berhasil mengklasifikasikan dengan benar sebanyak 48 sampel *Lumpy Skin* (TP) dan 135 sampel *Normal Skin* (TN), sementara terdapat 17 sampel *Lumpy Skin* yang salah diklasifikasikan sebagai *Normal Skin* (FN) dan 5 sampel *Normal Skin* yang salah diklasifikasikan sebagai *Lumpy Skin* (FP). Berikut adalah *confusion matrix* data validasi Skenario 1A.

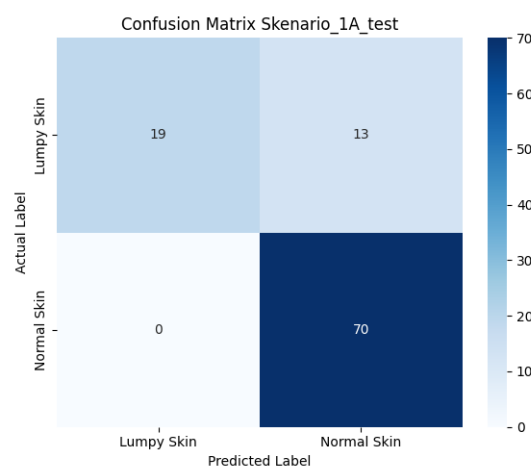


Gambar 4.2 *Confusion Matrix* Validasi Skenario 1A

Berdasarkan hasil tersebut, diperoleh nilai *precision* kelas *Lumpy Skin* sebesar 0,906, *recall* sebesar 0,738, dan F1-Score sebesar 0,813. Untuk kelas *Normal Skin*, diperoleh *precision* sebesar 0,888, *recall* sebesar 0,964, dan F1-Score sebesar 0,924. Secara keseluruhan, akurasi pada data validasi mencapai 89,27% dengan *macro average* F1-Score sebesar 0,869 dan *weighted average* F1-Score sebesar

0,893. Nilai *precision* kelas *Lumpy Skin* yang cukup tinggi (0,906) menunjukkan bahwa sebagian besar prediksi *Lumpy Skin* yang dikeluarkan model adalah benar, namun nilai *recall* yang lebih rendah (0,738) mengindikasikan model masih cukup sering melewati kasus *Lumpy Skin* yang sesungguhnya. Hal ini perlu menjadi perhatian mengingat dalam konteks medis-veteriner, kegagalan mendeteksi penyakit (*false negative*) berpotensi lebih merugikan dibandingkan *false positive*.

Evaluasi selanjutnya dilakukan pada data uji (*test set*) yang bersifat tetap (*fixed test set*) sebagaimana ditampilkan pada Gambar 4.x. Dari total 102 sampel uji yang terdiri dari 32 sampel *Lumpy Skin* dan 70 sampel *Normal Skin*, model mengklasifikasikan dengan benar 19 sampel *Lumpy Skin* (TP) dan 70 sampel *Normal Skin* (TN), sementara 13 sampel *Lumpy Skin* salah diklasifikasikan sebagai *Normal Skin* (FN) dan tidak ada sampel *Normal Skin* yang salah diklasifikasikan sebagai *Lumpy Skin* (FP = 0). Berikut adalah *confusion matrix* data uji Skenario 1A.



Gambar 4.3 *Confusion Matrix* Uji Skenario 1A

Berdasarkan hasil tersebut, diperoleh nilai *precision* kelas *Lumpy Skin* sebesar 1,000, *recall* sebesar 0,594, dan F1-Score sebesar 0,745. Untuk kelas *Normal Skin*, diperoleh *precision* sebesar 0,843, *recall* sebesar 1,000, dan F1-Score sebesar

0,915. Secara keseluruhan, akurasi pada data uji mencapai 87,25% dengan *macro average* F1-Score sebesar 0,830 dan *weighted average* F1-Score sebesar 0,873. Nilai *precision* kelas *Lumpy Skin* yang mencapai 1,000 menunjukkan bahwa seluruh prediksi *Lumpy Skin* yang dikeluarkan model adalah benar, yakni tidak ada satu pun sampel *Normal Skin* yang terprediksi sebagai *Lumpy Skin*. Namun nilai *recall* yang hanya mencapai 0,594 mengindikasikan bahwa model hanya mampu mendeteksi sekitar 59,4% dari keseluruhan kasus *Lumpy Skin* yang sesungguhnya, mencerminkan kecenderungan model untuk lebih konservatif dalam memprediksi kelas *Lumpy Skin* sehingga meminimalkan *false positive* namun dengan konsekuensi meningkatnya *false negative*. Secara keseluruhan, Skenario 1A dengan arsitektur VGG16 dan pembagian data 70:20:10 menunjukkan performa yang cukup baik namun belum optimal, terutama dalam hal kemampuan deteksi kelas minoritas (*Lumpy Skin*) pada data uji yang masih perlu ditingkatkan pada skenario-skenario berikutnya.

4.2.2 Uji Coba Skenario 1B

Skenario 1B merupakan skenario kedua yang menggunakan arsitektur VGG19 dengan rasio pembagian data yang sama seperti Skenario 1A, yaitu 70:20:10. Penggunaan VGG19 pada skenario ini bertujuan untuk membandingkan performa arsitektur yang lebih dalam dibandingkan VGG16, dengan jumlah lapisan konvolusi yang lebih banyak.

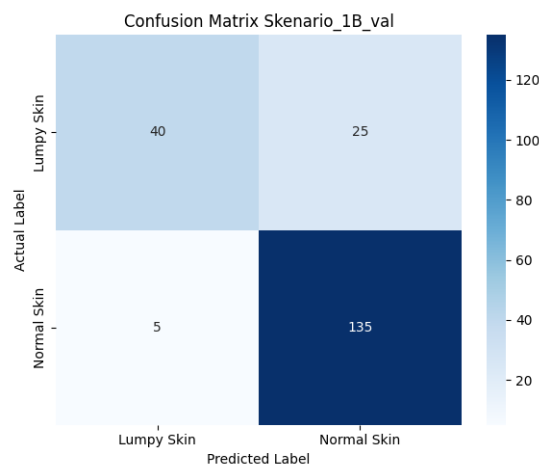


Gambar 4.4 Kurva *Accuracy* dan *Loss* Skenario 1B

Berdasarkan grafik *training history* pada Gambar 4.4, proses pelatihan Skenario 1B berlangsung selama 25 epoch. Pada epoch pertama, akurasi *training* dimulai pada nilai 70,99%, kemudian meningkat secara bertahap dan mencapai nilai tertinggi 90,79% pada epoch ke-22, sebelum sedikit turun menjadi 89,54% pada epoch ke-25. Nilai akurasi validasi (*val_accuracy*) menunjukkan fluktuasi yang lebih besar dibandingkan Skenario 1A, dengan nilai awal 72,20% pada epoch pertama yang kemudian melonjak ke 86,34% pada epoch kedua. Nilai tertinggi akurasi validasi mencapai 90,24% pada epoch ke-12, namun tidak bertahan konsisten hingga akhir dan turun menjadi 85,37% pada epoch ke-25. Pada grafik *loss*, nilai *training loss* menurun secara konsisten dari 0,6244 pada epoch pertama hingga 0,2538 pada epoch terakhir. Nilai *validation loss* menunjukkan fluktuasi yang cukup signifikan sepanjang proses pelatihan, dengan lonjakan yang mencolok pada epoch ke-16 mencapai 0,4852, sebelum kembali turun pada epoch-epoch berikutnya dan berakhir di angka 0,3633 pada epoch ke-25. Fluktuasi *validation loss* yang lebih besar ini, disertai gap yang semakin melebar antara *training loss* dan *validation loss* menjelang akhir pelatihan, mengindikasikan bahwa model VGG19 pada skenario ini mengalami kecenderungan *overfitting* yang lebih jelas

dibandingkan Skenario 1A, kemungkinan disebabkan oleh kapasitas model yang lebih besar yang tidak sepenuhnya termanfaatkan dengan jumlah data yang tersedia.

Evaluasi terhadap data validasi menghasilkan *confusion matrix* sebagaimana ditampilkan pada Gambar 4.5. Dari total 205 sampel validasi yang terdiri dari 65 sampel *Lumpy Skin* dan 140 sampel *Normal Skin*, model berhasil mengklasifikasikan dengan benar sebanyak 40 sampel *Lumpy Skin* (TP) dan 135 sampel *Normal Skin* (TN), sementara terdapat 25 sampel *Lumpy Skin* yang salah diklasifikasikan sebagai *Normal Skin* (FN) dan 5 sampel *Normal Skin* yang salah diklasifikasikan sebagai *Lumpy Skin* (FP). Berikut adalah *confusion matrix* data validasi Skenario 1B.

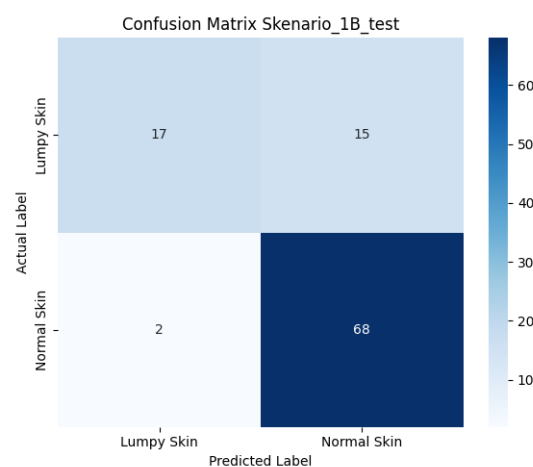


Gambar 4.5 *Confusion Matrix* Validasi Skenario 1B

Berdasarkan hasil tersebut, diperoleh nilai *precision* kelas *Lumpy Skin* sebesar 0,889, *recall* sebesar 0,615, dan F1-Score sebesar 0,727. Untuk kelas *Normal Skin*, diperoleh *precision* sebesar 0,844, *recall* sebesar 0,964, dan F1-Score sebesar 0,900. Secara keseluruhan, akurasi pada data validasi mencapai 85,37% dengan *macro average* F1-Score sebesar 0,814 dan *weighted average* F1-Score sebesar 0,847. Dibandingkan dengan Skenario 1A, akurasi validasi Skenario 1B lebih

rendah sebesar 3,90 poin persentase, dan nilai *recall* kelas *Lumpy Skin* juga lebih rendah (0,615 vs 0,738), menunjukkan bahwa VGG19 dengan pembagian data 70:20:10 menghasilkan kemampuan deteksi kelas *Lumpy Skin* yang lebih lemah pada data validasi.

Evaluasi selanjutnya dilakukan pada data uji (*fixed test set*) sebagaimana ditampilkan pada Gambar 4.6. Dari total 102 sampel uji yang terdiri dari 32 sampel *Lumpy Skin* dan 70 sampel *Normal Skin*, model mengklasifikasikan dengan benar 17 sampel *Lumpy Skin* (TP) dan 68 sampel *Normal Skin* (TN), sementara 15 sampel *Lumpy Skin* salah diklasifikasikan sebagai *Normal Skin* (FN) dan 2 sampel *Normal Skin* salah diklasifikasikan sebagai *Lumpy Skin* (FP). Berikut adalah *confusion matrix* data uji Skenario 1B.



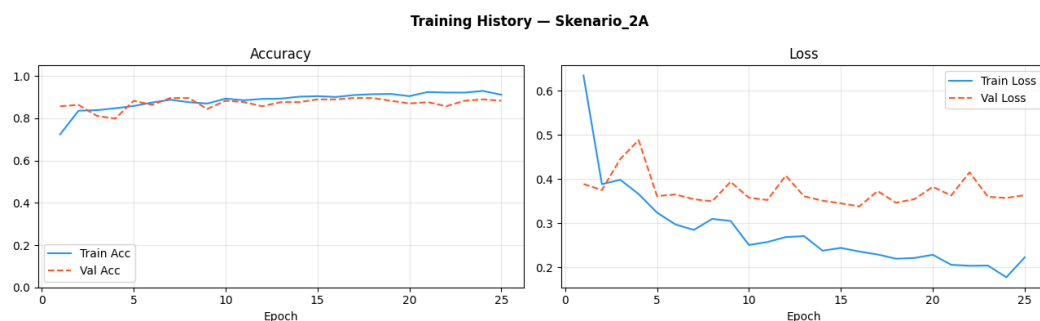
Gambar 4.6 *Confusion Matrix* Uji Skenario 1B

Berdasarkan hasil tersebut, diperoleh nilai *precision* kelas *Lumpy Skin* sebesar 0,895, *recall* sebesar 0,531, dan F1-Score sebesar 0,667. Untuk kelas *Normal Skin*, diperoleh *precision* sebesar 0,819, *recall* sebesar 0,971, dan F1-Score sebesar 0,889. Secara keseluruhan, akurasi pada data uji mencapai 83,33% dengan *macro average* F1-Score sebesar 0,778 dan *weighted average* F1-Score sebesar 0,824.

Apabila dibandingkan dengan Skenario 1A yang memperoleh akurasi uji 87,25%, Skenario 1B menunjukkan penurunan performa sebesar 3,92 poin persentase. Nilai *recall* kelas *Lumpy Skin* pada data uji yang hanya mencapai 0,531 menandakan bahwa model VGG19 pada rasio pembagian 70:20:10 lebih sering gagal mendeteksi kasus *Lumpy Skin* yang sesungguhnya dibandingkan VGG16. Secara keseluruhan, hasil Skenario 1B mengindikasikan bahwa penambahan kedalaman arsitektur dari VGG16 ke VGG19 tidak serta-merta meningkatkan performa klasifikasi pada kondisi pembagian data 70:20:10, dan justru menunjukkan kecenderungan *overfitting* yang lebih besar dengan performa akhir yang lebih rendah pada kedua set evaluasi.

4.2.3 Uji Coba Skenario 2A

Skenario 2A merupakan skenario ketiga yang kembali menggunakan arsitektur VGG16, namun dengan rasio pembagian data yang berbeda yaitu 75:15:10, terdiri dari 75% data latih, 15% data validasi, dan 10% data uji. Perubahan rasio ini bertujuan untuk mengevaluasi pengaruh penambahan proporsi data latih terhadap kemampuan generalisasi model.

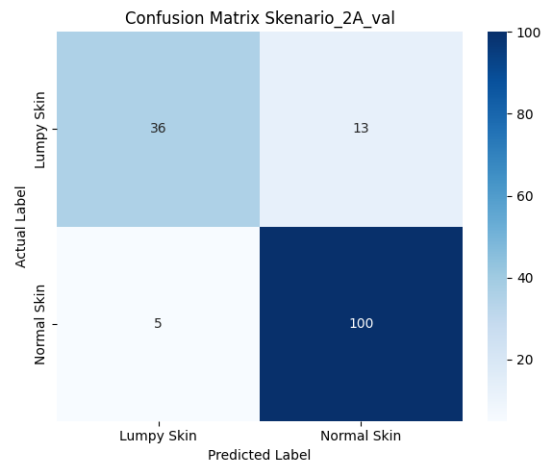


Gambar 4.7 Kurva *Accuracy* dan *Loss* Skenario 2A

Berdasarkan grafik *training history* pada Gambar 4.7, proses pelatihan Skenario 2A berlangsung selama 25 epoch. Akurasi *training* dimulai pada nilai 72,40% di epoch pertama, kemudian meningkat secara konsisten dan mencapai nilai tertinggi 92,97% pada epoch ke-24, sebelum sedikit turun menjadi 91,15% pada epoch ke-25. Nilai akurasi validasi (*val_accuracy*) menunjukkan tren yang relatif stabil dengan fluktuasi moderat, dimulai dari 85,71% pada epoch pertama dan mencapai nilai tertinggi 89,61% yang tercatat pada epoch ke-7, ke-8, ke-17, dan ke-18. Nilai akurasi validasi pada epoch terakhir berada di angka 88,31%. Pada grafik *loss*, nilai *training loss* menurun secara konsisten dari 0,6341 pada epoch pertama hingga 0,2227 pada epoch ke-25, dengan nilai terendah 0,1776 tercapai pada epoch ke-24. Nilai *validation loss* bergerak dalam rentang yang lebih tinggi dengan fluktuasi yang terlihat sepanjang proses pelatihan, dimulai dari 0,3887 dan berakhir di 0,3635 pada epoch terakhir, dengan lonjakan yang sempat terjadi pada epoch ke-4 mencapai 0,4879. Gap yang terbentuk antara *training loss* dan *validation loss* semakin melebar menjelang akhir pelatihan, namun akurasi validasi yang tetap terjaga di atas 88% menunjukkan bahwa model masih mampu melakukan generalisasi dengan baik meskipun terdapat indikasi *overfitting* ringan.

Evaluasi terhadap data validasi menghasilkan *confusion matrix* sebagaimana ditampilkan pada Gambar 4.8. Dari total 154 sampel validasi yang terdiri dari 49 sampel *Lumpy Skin* dan 105 sampel *Normal Skin*, model berhasil mengklasifikasikan dengan benar sebanyak 36 sampel *Lumpy Skin* (TP) dan 100 sampel *Normal Skin* (TN), sementara terdapat 13 sampel *Lumpy Skin* yang salah diklasifikasikan sebagai *Normal Skin* (FN) dan 5 sampel *Normal Skin* yang salah

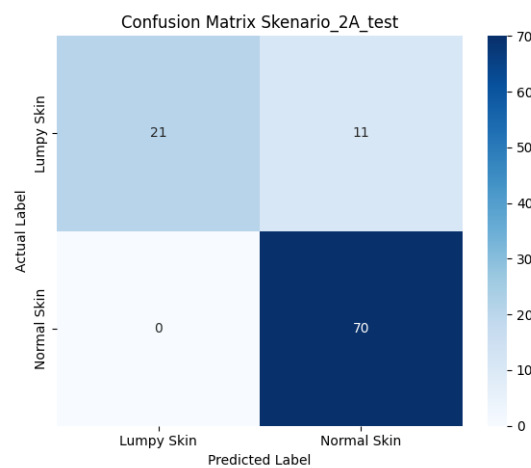
diklasifikasikan sebagai *Lumpy Skin* (FP). Berikut adalah *confusion matrix* data validasi Skenario 2A.



Gambar 4.8 *Confusion Matrix* Validasi Skenario 2A

Berdasarkan hasil tersebut, diperoleh nilai *precision* kelas *Lumpy Skin* sebesar 0,878, *recall* sebesar 0,735, dan F1-Score sebesar 0,800. Untuk kelas *Normal Skin*, diperoleh *precision* sebesar 0,885, *recall* sebesar 0,952, dan F1-Score sebesar 0,917. Secara keseluruhan, akurasi pada data validasi mencapai 88,31% dengan *macro average* F1-Score sebesar 0,859 dan *weighted average* F1-Score sebesar 0,880. Dibandingkan Skenario 1A dengan pembagian 70:20:10, akurasi validasi Skenario 2A lebih rendah sekitar 0,96 poin persentase, namun hal ini wajar mengingat jumlah sampel validasi yang lebih sedikit (154 vs 205 sampel) akibat proporsi data validasi yang diperkecil dari 20% menjadi 15%. Nilai *recall* kelas *Lumpy Skin* sebesar 0,735 menunjukkan kemampuan deteksi yang relatif sebanding dengan Skenario 1A (0,738), mengindikasikan bahwa penambahan data latih belum memberikan peningkatan yang signifikan pada kemampuan model mendeteksi kelas *Lumpy Skin* di data validasi.

Evaluasi selanjutnya dilakukan pada data uji (*fixed test set*) sebagaimana ditampilkan pada Gambar 4.9. Dari total 102 sampel uji yang terdiri dari 32 sampel *Lumpy Skin* dan 70 sampel *Normal Skin*, model mengklasifikasikan dengan benar 21 sampel *Lumpy Skin* (TP) dan 70 sampel *Normal Skin* (TN), sementara 11 sampel *Lumpy Skin* salah diklasifikasikan sebagai *Normal Skin* (FN) dan tidak ada sampel *Normal Skin* yang salah diklasifikasikan sebagai *Lumpy Skin* (FP = 0). Berikut adalah *confusion matrix* data uji Skenario 2A.



Gambar 4.9 *Confusion Matrix* Uji Skenario 2A

Berdasarkan hasil tersebut, diperoleh nilai *precision* kelas *Lumpy Skin* sebesar 1,000, *recall* sebesar 0,656, dan F1-Score sebesar 0,792. Untuk kelas *Normal Skin*, diperoleh *precision* sebesar 0,864, *recall* sebesar 1,000, dan F1-Score sebesar 0,927. Secara keseluruhan, akurasi pada data uji mencapai 89,22% dengan *macro average* F1-Score sebesar 0,860 dan *weighted average* F1-Score sebesar 0,893. Hasil ini menunjukkan peningkatan yang cukup berarti dibandingkan Skenario 1A yang memperoleh akurasi uji 87,25%, dengan selisih sebesar 1,97 poin persentase. Lebih signifikan lagi, nilai *recall* kelas *Lumpy Skin* pada data uji meningkat dari 0,594 (Skenario 1A) menjadi 0,656 (Skenario 2A), yang berarti model berhasil

mendeteksi 2 sampel *Lumpy Skin* tambahan secara benar. Nilai *precision* kelas *Lumpy Skin* yang kembali mencapai 1,000 menunjukkan konsistensi model dalam menghindari *false positive* pada kelas *Lumpy Skin*. Secara keseluruhan, Skenario 2A mengindikasikan bahwa penambahan proporsi data latih dari 70% menjadi 75% memberikan dampak positif terhadap performa model VGG16 pada data uji, meskipun peningkatan tersebut belum cukup besar untuk mengatasi kelemahan utama berupa masih tingginya angka *false negative* pada kelas *Lumpy Skin*.

4.2.4 Uji Coba Skenario 2B

Skenario 2B merupakan skenario keempat yang menggunakan arsitektur VGG19 dengan rasio pembagian data 75:15:10, yaitu 75% data latih, 15% data validasi, dan 10% data uji. Skenario ini merupakan pasangan komparasi dari Skenario 2A untuk mengevaluasi apakah arsitektur VGG19 yang lebih dalam mampu memberikan performa lebih baik dibandingkan VGG16 pada proporsi data latih yang sama.

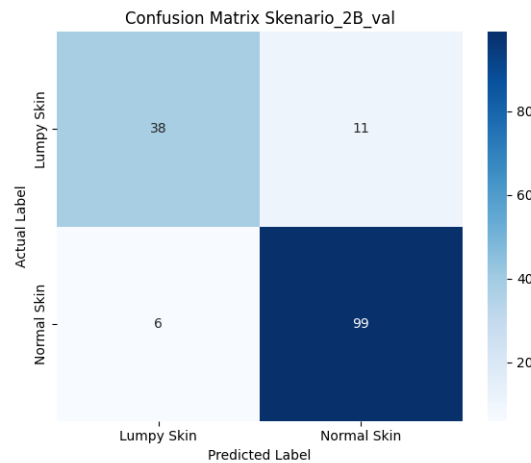


Gambar 4.10 Kurva *Accuracy* dan *Loss* Skenario 2B

Berdasarkan grafik *training history* pada Gambar 4.10, proses pelatihan Skenario 2B berlangsung selama 25 epoch. Akurasi *training* dimulai pada nilai 70,31% di epoch pertama, kemudian meningkat secara bertahap dan mencapai nilai

tertinggi 92,19% pada epoch ke-25. Nilai akurasi validasi (*val_accuracy*) menunjukkan tren yang relatif stabil dengan fluktuasi yang lebih terkendali dibandingkan Skenario 1B, dimulai dari 83,77% pada epoch pertama dan mencapai nilai tertinggi 90,26% yang tercatat pada epoch ke-9, ke-11, ke-16, dan ke-19. Nilai akurasi validasi pada epoch terakhir berada di angka 88,96%. Pada grafik *loss*, nilai *training loss* menurun secara konsisten dari 0,6195 pada epoch pertama hingga 0,2284 pada epoch ke-25. Nilai *validation loss* bergerak dalam rentang yang lebih tinggi dengan fluktuasi moderat sepanjang proses pelatihan, dimulai dari 0,3849 dan berakhir di 0,3259 pada epoch terakhir, dengan lonjakan yang sempat terjadi pada epoch ke-17 mencapai 0,4047. Meskipun gap antara *training loss* dan *validation loss* semakin melebar menjelang akhir pelatihan yang mengindikasikan *overfitting* ringan, pola kurva Skenario 2B secara keseluruhan lebih stabil dibandingkan Skenario 1B, menunjukkan bahwa penambahan proporsi data latih dari 70% menjadi 75% memberikan efek regularisasi yang lebih baik pada arsitektur VGG19.

Evaluasi terhadap data validasi menghasilkan *confusion matrix* sebagaimana ditampilkan pada Gambar 4.11. Dari total 154 sampel validasi yang terdiri dari 49 sampel *Lumpy Skin* dan 105 sampel *Normal Skin*, model berhasil mengklasifikasikan dengan benar sebanyak 38 sampel *Lumpy Skin* (TP) dan 99 sampel *Normal Skin* (TN), sementara terdapat 11 sampel *Lumpy Skin* yang salah diklasifikasikan sebagai *Normal Skin* (FN) dan 6 sampel *Normal Skin* yang salah diklasifikasikan sebagai *Lumpy Skin* (FP).

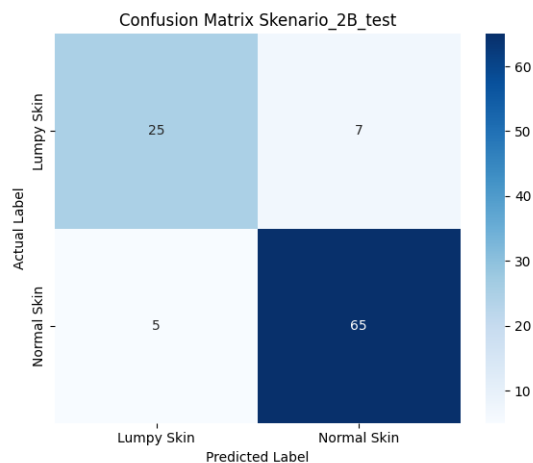


Gambar 4.11 *Confusion Matrix* Validasi Skenario 2B

Berdasarkan hasil tersebut, diperoleh nilai *precision* kelas *Lumpy Skin* sebesar 0,864, *recall* sebesar 0,776, dan F1-Score sebesar 0,817. Untuk kelas *Normal Skin*, diperoleh *precision* sebesar 0,900, *recall* sebesar 0,943, dan F1-Score sebesar 0,921. Secara keseluruhan, akurasi pada data validasi mencapai 88,96% dengan *macro average* F1-Score sebesar 0,869 dan *weighted average* F1-Score sebesar 0,888. Dibandingkan Skenario 2A (VGG16, 75:15:10) yang memperoleh akurasi validasi 88,31%, Skenario 2B menunjukkan peningkatan tipis sebesar 0,65 poin persentase. Yang lebih signifikan, nilai *recall* kelas *Lumpy Skin* pada Skenario 2B (0,776) lebih tinggi dibandingkan Skenario 2A (0,735), mengindikasikan bahwa VGG19 pada rasio 75:15:10 mampu mendeteksi lebih banyak kasus *Lumpy Skin* yang sesungguhnya pada data validasi, berbeda dengan pola yang ditemukan pada rasio 70:20:10 di mana VGG19 justru lebih lemah dari VGG16.

Evaluasi selanjutnya dilakukan pada data uji (*fixed test set*) sebagaimana ditampilkan pada Gambar 4.12. Dari total 102 sampel uji yang terdiri dari 32 sampel *Lumpy Skin* dan 70 sampel *Normal Skin*, model mengklasifikasikan dengan benar 25 sampel *Lumpy Skin* (TP) dan 65 sampel *Normal Skin* (TN), sementara 7 sampel

Lumpy Skin salah diklasifikasikan sebagai *Normal Skin* (FN) dan 5 sampel *Normal Skin* salah diklasifikasikan sebagai *Lumpy Skin* (FP).



Gambar 4.12 *Confusion Matrix* Uji Skenario 2B

Berdasarkan hasil tersebut, diperoleh nilai *precision* kelas *Lumpy Skin* sebesar 0,833, *recall* sebesar 0,781, dan F1-Score sebesar 0,806. Untuk kelas *Normal Skin*, diperoleh *precision* sebesar 0,903, *recall* sebesar 0,929, dan F1-Score sebesar 0,916. Secara keseluruhan, akurasi pada data uji mencapai 88,24% dengan *macro average* F1-Score sebesar 0,861 dan *weighted average* F1-Score sebesar 0,882. Meskipun akurasi data uji Skenario 2B (88,24%) sedikit lebih rendah dibandingkan Skenario 2A (89,22%), Skenario 2B justru menunjukkan keunggulan yang lebih berarti pada aspek keseimbangan klasifikasi. Nilai *recall* kelas *Lumpy Skin* sebesar 0,781 merupakan yang tertinggi di antara keempat skenario sebelumnya, menunjukkan bahwa model VGG19 dengan rasio 75:15:10 mampu mendeteksi 25 dari 32 sampel *Lumpy Skin* pada data uji. Meskipun demikian, kemunculan 5 *false positive* pada Skenario 2B menjadi catatan tersendiri, di mana *precision* kelas *Lumpy Skin* yang sebesar 0,833 lebih rendah dibandingkan beberapa skenario

sebelumnya yang mencapai 1,000. Secara keseluruhan, Skenario 2B menunjukkan performa model yang lebih seimbang antara *precision* dan *recall* untuk kelas *Lumpy Skin*, menjadikannya kandidat yang menarik apabila prioritas utama adalah meminimalkan kasus *false negative* dalam deteksi penyakit.

4.2.5 Uji Coba Skenario 3A

Skenario 3A merupakan skenario kelima yang menggunakan arsitektur VGG16 dengan rasio pembagian data 80:10:10, yaitu 80% data latih, 10% data validasi, dan 10% data uji. Rasio ini menempatkan proporsi data latih terbesar di antara seluruh skenario yang diujikan, dengan konsekuensi berkurangnya jumlah sampel validasi menjadi hanya 10% dari keseluruhan dataset. Pengujian ini bertujuan untuk mengevaluasi apakah penambahan data latih yang lebih agresif mampu mendorong peningkatan performa model secara lebih signifikan.

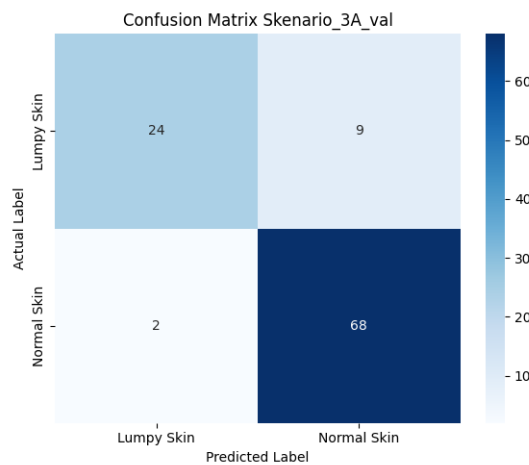


Gambar 4.13 Kurva *Accuracy* dan *Loss* Skenario 3A

Berdasarkan grafik *training history* pada Gambar 4.13, proses pelatihan Skenario 3A berlangsung selama 25 epoch. Akurasi *training* dimulai pada nilai 72,89% di epoch pertama, kemudian meningkat secara konsisten dan mencapai nilai tertinggi 92,67% pada epoch ke-24, sebelum sedikit turun menjadi 91,82% pada epoch ke-25. Nilai akurasi validasi (*val_accuracy*) menunjukkan fluktuasi

yang lebih besar dibandingkan skenario-skenario sebelumnya, dimulai dari 84,47% pada epoch pertama dengan nilai tertinggi 89,32% yang tercatat pada epoch ke-6, ke-23, dan ke-25. Fluktuasi yang lebih tinggi pada kurva validasi ini dapat dipahami sebagai konsekuensi dari sedikitnya jumlah sampel validasi (hanya 10% dari dataset), sehingga perubahan klasifikasi pada beberapa sampel saja sudah berdampak cukup besar terhadap nilai akurasi validasi. Pada grafik *loss*, nilai *training loss* menurun secara konsisten dari 0,5463 pada epoch pertama hingga 0,2006 pada epoch ke-25, dengan nilai terendah 0,1924 tercapai pada epoch ke-24. Nilai *validation loss* menunjukkan fluktuasi yang cukup mencolok sepanjang proses pelatihan, dengan lonjakan signifikan pada epoch ke-19 mencapai 0,4819 dan kembali terjadi pada epoch ke-24 mencapai 0,4393, sebelum turun menjadi 0,3775 pada epoch terakhir. Jarak yang lebar antara *training loss* dan *validation loss* sepanjang proses pelatihan, disertai fluktuasi *validation loss* yang tidak stabil, mengindikasikan bahwa pengurangan proporsi data validasi menjadi 10% menyebabkan estimasi generalisasi model menjadi kurang konsisten, meskipun akurasi validasi akhir tetap berada pada level yang kompetitif.

Evaluasi terhadap data validasi menghasilkan *confusion matrix* sebagaimana ditampilkan pada Gambar 4.14. Dari total 103 sampel validasi yang terdiri dari 33 sampel *Lumpy Skin* dan 70 sampel *Normal Skin*, model berhasil mengklasifikasikan dengan benar sebanyak 24 sampel *Lumpy Skin* (TP) dan 68 sampel *Normal Skin* (TN), sementara terdapat 9 sampel *Lumpy Skin* yang salah diklasifikasikan sebagai *Normal Skin* (FN) dan 2 sampel *Normal Skin* yang salah diklasifikasikan sebagai *Lumpy Skin* (FP).

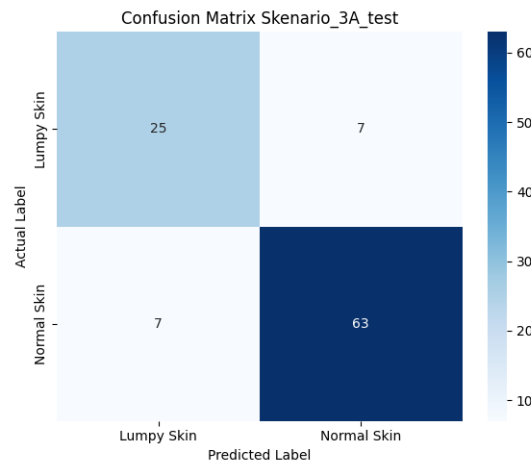


Gambar 4.14 *Confusion Matrix* Validasi Skenario 3A

Berdasarkan hasil tersebut, diperoleh nilai *precision* kelas *Lumpy Skin* sebesar 0,923, *recall* sebesar 0,727, dan F1-Score sebesar 0,814. Untuk kelas *Normal Skin*, diperoleh *precision* sebesar 0,883, *recall* sebesar 0,971, dan F1-Score sebesar 0,925. Secara keseluruhan, akurasi pada data validasi mencapai 89,32% dengan *macro average* F1-Score sebesar 0,870 dan *weighted average* F1-Score sebesar 0,891. Nilai akurasi validasi ini merupakan yang tertinggi di antara seluruh skenario VGG16 yang telah diujikan (1A: 89,27%; 2A: 88,31%; 3A: 89,32%), meskipun selisihnya sangat tipis. Nilai *precision* kelas *Lumpy Skin* sebesar 0,923 juga merupakan yang tertinggi di antara skenario VGG16, menunjukkan bahwa model dengan data latih yang lebih banyak cenderung lebih selektif dalam memprediksi kelas *Lumpy Skin*.

Evaluasi selanjutnya dilakukan pada data uji (*fixed test set*) sebagaimana ditampilkan pada Gambar 4.15. Dari total 102 sampel uji yang terdiri dari 32 sampel *Lumpy Skin* dan 70 sampel *Normal Skin*, model mengklasifikasikan dengan benar 25 sampel *Lumpy Skin* (TP) dan 63 sampel *Normal Skin* (TN), sementara 7 sampel

Lumpy Skin salah diklasifikasikan sebagai *Normal Skin* (FN) dan 7 sampel *Normal Skin* salah diklasifikasikan sebagai *Lumpy Skin* (FP).



Gambar 4.15 *Confusion Matrix* Uji Skenario 3A

Berdasarkan hasil tersebut, diperoleh nilai *precision* kelas *Lumpy Skin* sebesar 0,781, *recall* sebesar 0,781, dan F1-Score sebesar 0,781. Untuk kelas *Normal Skin*, diperoleh *precision* sebesar 0,900, *recall* sebesar 0,900, dan F1-Score sebesar 0,900. Secara keseluruhan, akurasi pada data uji mencapai 86,27% dengan *macro average* F1-Score sebesar 0,841 dan *weighted average* F1-Score sebesar 0,863. Skenario 3A menampilkan karakteristik yang berbeda dari skenario VGG16 sebelumnya, yakni nilai *precision* dan *recall* kelas *Lumpy Skin* yang seimbang sempurna pada angka 0,781 — model tidak lagi terlalu konservatif dalam memprediksi kelas *Lumpy Skin* (tidak ada lagi *precision* 1,000 dengan *recall* rendah), namun justru menunjukkan keseimbangan yang lebih baik antara kemampuan mendeteksi kasus positif dan menghindari prediksi positif yang salah. Meskipun akurasi data uji Skenario 3A (86,27%) lebih rendah dibandingkan Skenario 1A (87,25%) dan 2A (89,22%), nilai F1-Score kelas *Lumpy Skin* yang seimbang ini mengindikasikan bahwa model Skenario 3A memiliki profil performa

yang lebih proporsional. Secara keseluruhan, hasil Skenario 3A menunjukkan bahwa penambahan proporsi data latih hingga 80% membawa perubahan perilaku klasifikasi yang cukup berbeda, dengan implikasi praktis bahwa model ini lebih andal dalam mendeteksi *Lumpy Skin* secara konsisten meskipun dengan sedikit pengorbanan pada nilai akurasi keseluruhan.

4.2.6 Uji Coba Skenario 3B

Skenario 3B merupakan skenario keenam sekaligus terakhir dalam penelitian ini, menggunakan arsitektur VGG19 dengan rasio pembagian data 80:10:10, yaitu 80% data latih, 10% data validasi, dan 10% data uji. Skenario ini merupakan pasangan komparasi dari Skenario 3A untuk mengevaluasi performa arsitektur VGG19 pada proporsi data latih terbesar yang diujikan dalam penelitian ini.

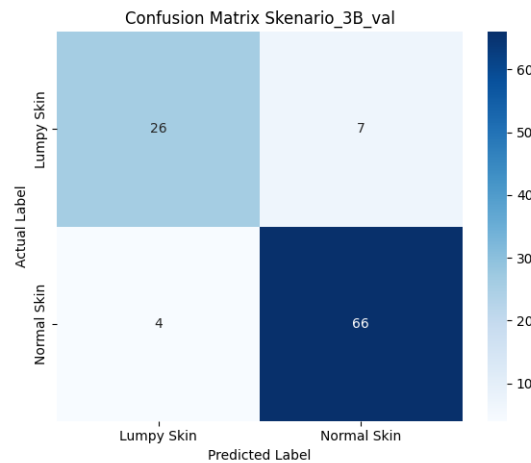


Gambar 4.16 Kurva *Accuracy* dan *Loss* Skenario 3B

Berdasarkan grafik *training history* pada Gambar 4.16, proses pelatihan Skenario 3B berlangsung selama 25 epoch. Akurasi *training* dimulai pada nilai yang paling rendah di antara seluruh skenario, yakni 66,91% pada epoch pertama, namun kemudian meningkat secara konsisten dan mencapai nilai tertinggi 90,96% pada epoch ke-24, sebelum sedikit turun menjadi 90,84% pada epoch ke-25. Nilai akurasi validasi (*val_accuracy*) menunjukkan fluktuasi yang cukup besar

sebagaimana juga terlihat pada Skenario 3A, dengan nilai awal 73,79% pada epoch pertama yang kemudian melonjak signifikan ke 87,38% pada epoch kedua. Nilai tertinggi akurasi validasi mencapai 90,29% yang tercatat pada beberapa epoch, antara lain epoch ke-7, ke-14, ke-15, ke-17, ke-20, ke-21, dan ke-24, sementara nilai akhir pada epoch ke-25 berada di angka 89,32%. Fluktuasi yang lebih besar pada kurva validasi, serupa dengan Skenario 3A, merupakan konsekuensi dari sedikitnya jumlah sampel validasi yang hanya berjumlah 103 sampel. Pada grafik *loss*, nilai *training loss* mengalami penurunan yang cukup dramatis dari 0,8102 pada epoch pertama — nilai awal tertinggi di antara seluruh skenario — hingga 0,2267 pada epoch ke-25. Nilai *validation loss* bergerak dalam rentang yang lebih tinggi dengan fluktuasi moderat, dimulai dari 0,5691 dan berakhir di 0,3360 pada epoch terakhir, dengan lonjakan yang sempat terjadi pada epoch ke-8 (0,4139) dan epoch ke-11 (0,4400). Secara keseluruhan, pola kurva Skenario 3B menunjukkan proses pembelajaran yang lebih dinamis dibandingkan Skenario 3A, dengan gap antara *training loss* dan *validation loss* yang tetap terjaga dalam rentang yang wajar menjelang akhir pelatihan.

Evaluasi terhadap data validasi menghasilkan *confusion matrix* sebagaimana ditampilkan pada Gambar 4.17. Dari total 103 sampel validasi yang terdiri dari 33 sampel *Lumpy Skin* dan 70 sampel *Normal Skin*, model berhasil mengklasifikasikan dengan benar sebanyak 26 sampel *Lumpy Skin* (TP) dan 66 sampel *Normal Skin* (TN), sementara terdapat 7 sampel *Lumpy Skin* yang salah diklasifikasikan sebagai *Normal Skin* (FN) dan 4 sampel *Normal Skin* yang salah diklasifikasikan sebagai *Lumpy Skin* (FP).

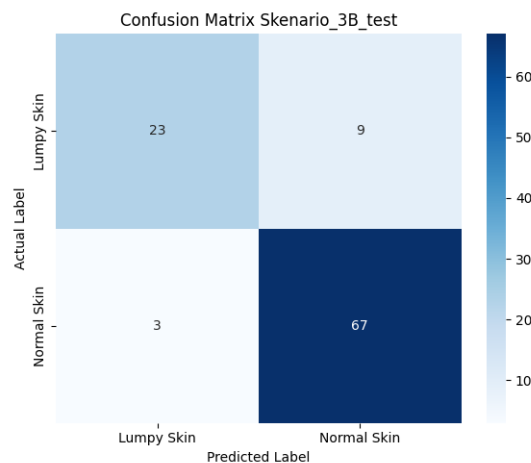


Gambar 4.17 *Confusion Matrix* Validasi Skenario 3B

Berdasarkan hasil tersebut, diperoleh nilai *precision* kelas *Lumpy Skin* sebesar 0,867, *recall* sebesar 0,788, dan F1-Score sebesar 0,825. Untuk kelas *Normal Skin*, diperoleh *precision* sebesar 0,904, *recall* sebesar 0,943, dan F1-Score sebesar 0,923. Secara keseluruhan, akurasi pada data validasi mencapai 89,32% dengan *macro average* F1-Score sebesar 0,874 dan *weighted average* F1-Score sebesar 0,892. Dibandingkan Skenario 3A (VGG16, 80:10:10) yang memperoleh akurasi validasi identik sebesar 89,32%, Skenario 3B menunjukkan keunggulan pada nilai *recall* kelas *Lumpy Skin* (0,788 vs 0,727) dan F1-Score kelas *Lumpy Skin* (0,825 vs 0,814), mengindikasikan bahwa VGG19 pada rasio 80:10:10 mampu mendeteksi lebih banyak kasus *Lumpy Skin* yang sesungguhnya pada data validasi dibandingkan VGG16, pola yang konsisten dengan temuan pada Skenario 2A dan 2B.

Evaluasi selanjutnya dilakukan pada data uji (*fixed test set*) sebagaimana ditampilkan pada Gambar 4.18. Dari total 102 sampel uji yang terdiri dari 32 sampel *Lumpy Skin* dan 70 sampel *Normal Skin*, model mengklasifikasikan dengan benar

23 sampel *Lumpy Skin* (TP) dan 67 sampel *Normal Skin* (TN), sementara 9 sampel *Lumpy Skin* salah diklasifikasikan sebagai *Normal Skin* (FN) dan 3 sampel *Normal Skin* salah diklasifikasikan sebagai *Lumpy Skin* (FP).



Gambar 4.18 *Confusion Matrix* Uji Skenario 3B

Berdasarkan hasil tersebut, diperoleh nilai *precision* kelas *Lumpy Skin* sebesar 0,885, *recall* sebesar 0,719, dan F1-Score sebesar 0,793. Untuk kelas *Normal Skin*, diperoleh *precision* sebesar 0,882, *recall* sebesar 0,957, dan F1-Score sebesar 0,918. Secara keseluruhan, akurasi pada data uji mencapai 88,24% dengan *macro average* F1-Score sebesar 0,856 dan *weighted average* F1-Score sebesar 0,879. Akurasi data uji Skenario 3B (88,24%) lebih tinggi dibandingkan Skenario 3A (86,27%), menunjukkan bahwa VGG19 dengan rasio 80:10:10 mampu menghasilkan generalisasi yang lebih baik pada data uji dibandingkan VGG16 pada rasio yang sama. Nilai *precision* kelas *Lumpy Skin* sebesar 0,885 yang lebih tinggi dibandingkan Skenario 3A (0,781) mengindikasikan bahwa Skenario 3B lebih selektif dalam memprediksi kelas *Lumpy Skin*, meskipun nilai *recall*-nya (0,719) lebih rendah dibandingkan Skenario 3A (0,781). Secara keseluruhan, Skenario 3B

menyelesaikan rangkaian enam skenario pengujian dengan performa yang kompetitif, dan bersama Skenario 2B menegaskan pola bahwa arsitektur VGG19 cenderung menunjukkan keseimbangan *precision-recall* yang lebih baik pada proporsi data latih yang lebih besar, meskipun tidak selalu unggul dalam hal akurasi keseluruhan dibandingkan VGG16 pada kondisi yang sama.

4.2.7 Hasil Uji Coba Skenario Terbaik

Berdasarkan evaluasi seluruh uji coba yang telah dilakukan terhadap keenam skenario, penetapan skenario terbaik dalam penelitian ini dilakukan dengan mempertimbangkan akurasi keseluruhan pada data uji sebagai kriteria utama, yang diperkuat oleh nilai *precision*, *recall*, dan F1-Score khususnya pada kelas *Lumpy Skin* sebagai kelas minoritas yang menjadi fokus utama klasifikasi. Perbandingan akurasi data uji seluruh skenario disajikan pada Tabel 4.3 berikut.

Tabel 4.3 Perbandingan Akurasi Data Uji Seluruh Skenario

Skenario	Arsitektur	Split Data	Akurasi	Mac F1
1A	VGG16	70:20:10	87,25%	0,830
1B	VGG19	70:20:10	83,33%	0,778
2A	VGG16	75:15:10	89,22%	0,860
2B	VGG19	75:15:10	88,24%	0,861
3A	VGG16	80:10:10	86,27%	0,841
3B	VGG19	80:10:10	88,24%	0,856

Berdasarkan tabel di atas, Skenario 2A dengan arsitektur VGG16 dan pembagian data 75:15:10 ditetapkan sebagai skenario terbaik dalam penelitian ini. Skenario 2A mencatat akurasi data uji tertinggi di antara seluruh skenario sebesar 89,22%, dengan nilai *precision* kelas *Lumpy Skin* sebesar 1,000 yang menunjukkan bahwa seluruh prediksi *Lumpy Skin* yang dikeluarkan model adalah benar tanpa satu pun *false positive*. Nilai F1-Score kelas *Lumpy Skin* sebesar 0,792 merupakan yang tertinggi di antara seluruh skenario VGG16, dan *macro average* F1-Score

sebesar 0,860 juga merupakan yang tertinggi secara keseluruhan di antara keenam skenario. Dibandingkan skenario lainnya, Skenario 2A menunjukkan kombinasi performa terbaik antara akurasi keseluruhan dan kemampuan klasifikasi kelas *Lumpy Skin*, menjadikannya model yang paling optimal untuk diintegrasikan sebagai sistem klasifikasi otomatis *Lumpy Skin Disease* pada sapi. Model terbaik Skenario 2A selanjutnya diimplementasikan ke dalam antarmuka web berbasis Gradio sebagai sistem klasifikasi *Lumpy Skin Disease* yang dapat digunakan secara praktis, di mana pengguna dapat mengunggah citra kulit sapi dan memperoleh hasil klasifikasi secara otomatis beserta nilai kepercayaan (*confidence score*) prediksi model.

4.3 Implementasi dan Pengujian Sistem Berbasis Web

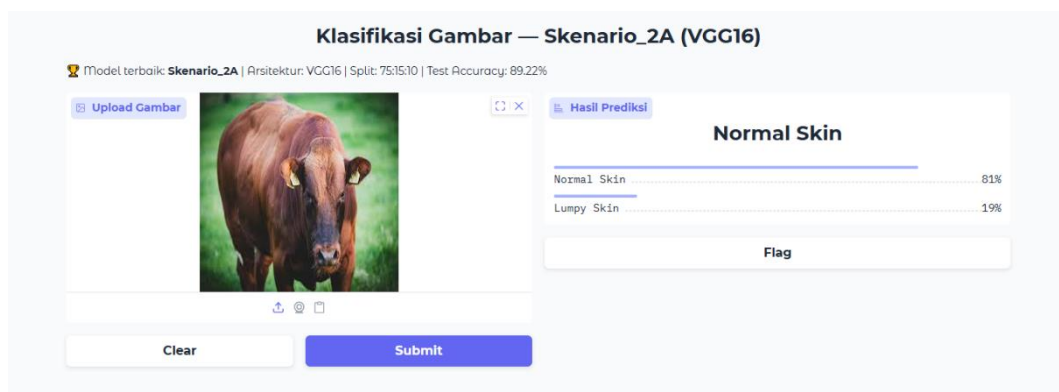
Sub-bab ini memaparkan implementasi model terbaik dari Skenario 2A ke dalam sistem deteksi berbasis web menggunakan Gradio, serta pengujian sistem yang telah dibangun. Implementasi ini bertujuan untuk membuktikan bahwa model yang telah dilatih dapat dioperasikan secara nyata dan memberikan hasil prediksi secara langsung oleh pengguna tanpa memerlukan keahlian teknis khusus. Pembahasan mencakup implementasi antarmuka aplikasi dan pengujian sistem menggunakan citra uji.

4.3.1 Implementasi Antarmuka Sistem

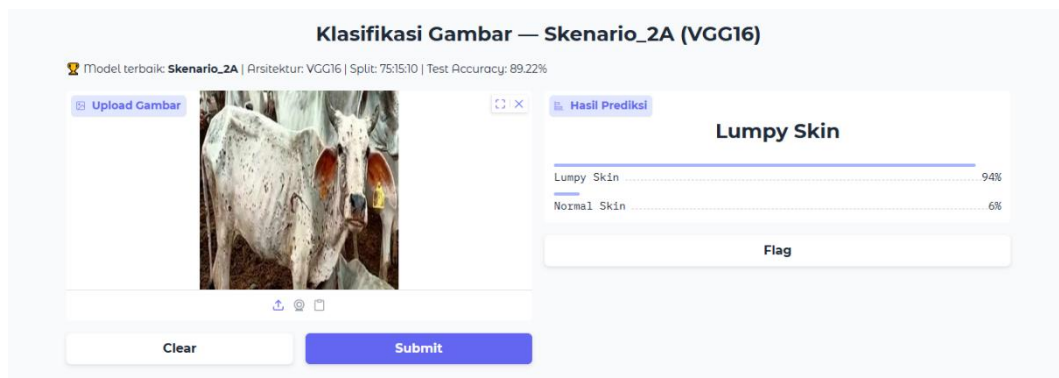
Nilai akurasi yang dihasilkan selama proses pelatihan dan evaluasi model perlu dibuktikan secara nyata melalui sebuah sistem yang dapat digunakan langsung oleh pengguna. Oleh karena itu, model terbaik pada Skenario 2A yang

disimpan dalam format file model (.h5) akan diimplementasikan ke dalam antarmuka web interaktif Gradio. Gradio dipilih karena kemampuannya dalam membangun antarmuka yang sederhana namun fungsional hanya dengan beberapa baris kode Python, sehingga memudahkan proses pengembangan dan pengujian sistem secara cepat.

Antarmuka dirancang agar pengguna dapat mengoperasikan sistem dengan mudah, meskipun tidak memiliki keahlian di bidang kecerdasan buatan. Pengguna cukup mengunggah citra kulit sapi melalui tombol yang tersedia, selanjutnya sistem akan mengolah citra secara otomatis menggunakan model yang telah dilatih dan menampilkan hasil prediksinya berupa nama kelas (*Lumpy Skin* atau *Normal Skin*) beserta nilai persentase kepercayaan (*confidence score*) untuk masing-masing kelas. Proses implementasi antarmuka ketika pengguna mengunggah dan mengirimkan citra ke dalam sistem diilustrasikan pada Gambar 4.19 dan Gambar 4.20 berikut.



Gambar 4.19 Tampilan sistem dengan hasil prediksi kelas Normal Skin



Gambar 4.20 Tampilan sistem dengan hasil prediksi kelas Lumpy Skin

Antarmuka aplikasi Sistem Deteksi *Lumpy Skin Disease* (LSD) yang dibangun menggunakan Gradio terdiri dari dua panel utama yang tersusun secara berdampingan. Panel sebelah kiri berfungsi sebagai area masukan (*input*) dengan label "Unggah Citra Kulit Sapi", di mana pengguna dapat mengunggah citra kulit sapi yang ingin dideteksi. Di bawah panel input terdapat tombol "Submit" untuk mengirimkan citra ke sistem dan tombol "Clear" untuk menghapus citra yang telah diunggah. Panel sebelah kanan berfungsi sebagai area keluaran (*output*) dengan label "Hasil Prediksi", yang menampilkan nama kelas hasil prediksi secara menonjol di bagian atas beserta diagram batang persentase *confidence score* untuk masing-masing kelas secara berurutan.

Berdasarkan Gambar 4.19, ketika sistem diberikan citra kulit sapi yang sehat tanpa gejala LSD, model memprediksi kelas Normal Skin dengan tingkat kepercayaan sebesar 81% dan probabilitas kelas *Lumpy Skin* sebesar 19%. Nilai *confidence score* yang tidak terlalu tinggi ini mengindikasikan bahwa model sedikit lebih ragu dalam mengklasifikasikan citra kulit normal, yang kemungkinan disebabkan oleh kondisi citra uji yang berbeda dari karakteristik data latih, seperti perbedaan sudut pengambilan gambar, latar belakang citra, maupun kondisi

pencahayaannya. Meskipun demikian, prediksi yang dihasilkan tetap benar sesuai dengan kondisi aktual citra yang diujikan.

Sementara itu, pada Gambar 4.20, ketika sistem diberikan citra kulit sapi yang menunjukkan gejala LSD, model berhasil memprediksi kelas *Lumpy Skin* dengan tingkat kepercayaan yang sangat tinggi yaitu 94%, sementara probabilitas kelas *Normal Skin* hanya sebesar 6%. Tingginya nilai *confidence score* ini menunjukkan bahwa model mempunyai keyakinan yang sangat kuat dalam mengenali ciri-ciri visual LSD pada citra tersebut, seperti adanya benjolan dan lesi yang terlihat jelas pada permukaan kulit sapi.

4.3.2 Pengujian Sistem

Guna memverifikasi bahwa model yang telah diimplementasikan ke dalam Gradio berfungsi sebagaimana mestinya, sistem di uji menggunakan 10 citra uji yang terdiri dari 5 citra kulit normal (*Normal Skin*) dan 5 citra kulit sapi yang terindikasi LSD (*Lumpy Skin*). Citra-citra uji yang digunakan merupakan citra yang tidak termasuk dalam dataset pelatihan maupun validasi, sehingga hasil pengujian ini dapat mencerminkan kemampuan generalisasi model secara lebih objektif terhadap data yang benar-benar baru. Setiap citra diunggah satu per satu melalui antarmuka aplikasi, kemudian sistem akan menghasilkan prediksi kelas beserta nilai *confidence score* untuk tiap kelas. Hasil prediksi yang diperoleh kemudian dibandingkan dengan label aktual citra untuk menentukan apakah prediksi tersebut benar (*correct*) atau salah (*incorrect*). Seluruh hasil pengujian dirangkum secara terstruktur pada Tabel 4.4 berikut.

Tabel 4.4 Hasil Pengujian Sistem

No	Citra Uji	Kondisi Sebenarnya	Hasil Prediksi	Status
1		Lumpy Skin	Lumpy Skin : 94% Normal Skin : 6%	Sesuai
2		Lumpy Skin	Lumpy Skin : 96% Normal Skin : 4%	Sesuai
3		Lumpy Skin	Lumpy Skin : 78% Normal Skin : 22%	Sesuai
4		Lumpy Skin	Lumpy Skin : 61% Normal Skin : 39%	Sesuai
5		Lumpy Skin	Lumpy Skin : 99% Normal Skin : 1%	Sesuai
6		Normal Skin	Lumpy Skin : 28% Normal Skin : 72%	Sesuai

7		Normal Skin	Lumpy Skin : 1% Normal Skin : 99%	Sesuai
8		Normal Skin	Lumpy Skin : 33% Normal Skin : 67%	Sesuai
9		Normal Skin	Lumpy Skin : 28% Normal Skin : 72%	Sesuai
10		Normal Skin	Lumpy Skin : 19% Normal Skin : 81%	Sesuai

Berdasarkan Tabel 4.4, sistem berhasil mengklasifikasikan seluruh 10 citra uji dengan benar. Pada kelima citra *Lumpy Skin*, model menunjukkan *confidence score* yang bervariasi dalam memprediksi kelas *Lumpy Skin*, dengan nilai tertinggi sebesar 99% pada citra ke-5 dan nilai terendah sebesar 61% pada citra ke-4. Variasi nilai kepercayaan ini dapat dipahami sebagai cerminan dari tingkat kemiripan visual antar citra, di mana citra dengan gejala lesi *Lumpy Skin* yang lebih jelas dan dominan cenderung menghasilkan nilai kepercayaan yang lebih tinggi, sementara citra dengan tampilan lesi yang lebih samar atau disertai variasi warna kulit yang kompleks menghasilkan nilai kepercayaan yang lebih rendah meskipun tetap

diklasifikasikan dengan benar. Rata-rata nilai kepercayaan prediksi kelas *Lumpy Skin* pada kelima citra berlabel *Lumpy Skin* mencapai 85,6%, menunjukkan bahwa model secara umum mampu mengenali pola visual *Lumpy Skin Disease* dengan tingkat keyakinan yang cukup tinggi.

Pada kelompok citra berlabel *Normal Skin*, model menunjukkan performa yang sangat baik dengan nilai kepercayaan prediksi kelas *Normal Skin* berkisar antara 67% hingga 99%. Nilai kepercayaan tertinggi sebesar 99% diperoleh pada citra ke-7, sementara nilai terendah sebesar 67% pada citra ke-8 yang mengindikasikan bahwa citra tersebut memiliki karakteristik visual yang sedikit lebih ambigu namun tetap berhasil diklasifikasikan dengan benar. Rata-rata nilai kepercayaan prediksi kelas *Normal Skin* pada kelima citra berlabel *Normal Skin* mencapai 78,2%, yang meskipun sedikit lebih rendah dibandingkan kelompok *Lumpy Skin*, tetap menunjukkan kemampuan diskriminasi model yang memadai. Secara keseluruhan, hasil pengujian sistem membuktikan bahwa model terbaik Skenario 2A yang diimplementasikan pada antarmuka Gradio mampu melakukan klasifikasi *Lumpy Skin Disease* secara akurat dan konsisten pada sampel citra baru, dengan tingkat kepercayaan prediksi yang cukup tinggi pada kedua kelas, sehingga layak untuk digunakan sebagai alat bantu identifikasi awal *Lumpy Skin Disease* pada sapi secara praktis di lapangan.

4.4 Pembahasan

Bagian ini menyajikan pembahasan menyeluruh dan perbandingan hasil dari keenam skenario yang telah diuji dalam penelitian ini. Perbandingan dilakukan berdasarkan beberapa aspek, yaitu akurasi keseluruhan, *precision*, *recall*, dan *F1-*

score pada masing-masing kelas. Rekapitulasi hasil seluruh skenario pada data validasi dan data uji disajikan pada Tabel 4.5 dan Tabel 4.6 berikut.

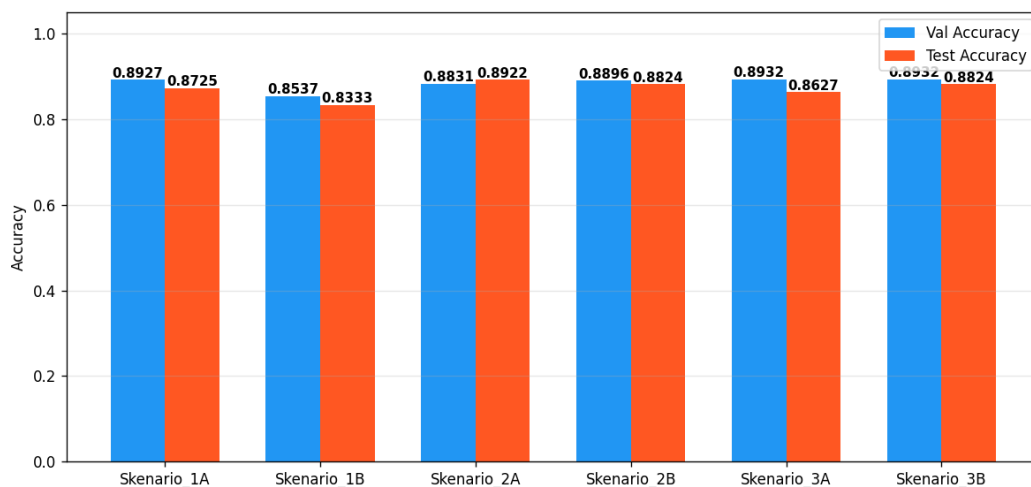
Tabel 4.5 Rekapitulasi Hasil Data Validasi Seluruh Skenario

Skenario	Akurasi	Precision Lumpy	Recall Lumpy	F1 Lumpy	F1 Normal	Macro F1
1A	89,27%	0,906	0,738	0,813	0,924	0,869
1B	85,37%	0,889	0,615	0,727	0,900	0,814
2A	88,31%	0,878	0,735	0,800	0,917	0,859
2B	88,96%	0,864	0,776	0,817	0,921	0,869
3A	89,32%	0,923	0,727	0,814	0,925	0,870
3B	89,32%	0,867	0,788	0,825	0,923	0,874

Tabel 4.6 Rekapitulasi Hasil Data Uji Seluruh Skenario

Skenario	Akurasi	Precision Lumpy	Recall Lumpy	F1 Lumpy	F1 Normal	Macro F1
1A	87,25%	1,000	0,594	0,745	0,915	0,830
1B	83,33%	0,895	0,531	0,667	0,889	0,778
2A	89,22%	1,000	0,656	0,792	0,927	0,860
2B	88,24%	0,833	0,781	0,806	0,916	0,861
3A	86,27%	0,781	0,781	0,781	0,900	0,841
3B	88,24%	0,885	0,719	0,793	0,918	0,856

Gambaran perbandingan akurasi validasi dan akurasi uji seluruh skenario secara visual disajikan pada Gambar 4.21. Berdasarkan grafik tersebut, terlihat bahwa akurasi validasi seluruh skenario berada dalam rentang yang relatif berdekatan antara 0,8537 hingga 0,8932, sementara akurasi data uji menunjukkan variasi yang lebih besar dengan rentang antara 0,8333- 0,8922. Pola yang konsisten terlihat pada seluruh skenario bahwa akurasi validasi cenderung lebih tinggi dibandingkan akurasi data uji pada skenario yang sama, yang mengindikasikan adanya perbedaan karakteristik distribusi antara data validasi dan data uji meskipun keduanya diambil dari dataset yang sama melalui pembagian stratifikasi.



Gambar 4.21 Perbandingan Akurasi Data Validasi dan Uji Seluruh Skenario

4.4.1 Pengaruh Proporsi Data

Berdasarkan hasil evaluasi keenam skenario yang telah dijabarkan, proporsi pembagian data memberikan pengaruh yang cukup signifikan terhadap performa model klasifikasi *Lumpy Skin Disease*. Secara umum, terdapat pola yang dapat diamati bahwa penambahan proporsi data latih tidak selalu berbanding lurus dengan peningkatan akurasi keseluruhan, namun memberikan dampak yang berbeda terhadap masing-masing arsitektur yang digunakan.

Pada kelompok skenario dengan arsitektur VGG16, rasio 75:15:10 (Skenario 2A) menghasilkan akurasi data uji tertinggi sebesar 89,22%, melampaui rasio 70:20:10 (Skenario 1A, 87,25%) maupun rasio 80:10:10 (Skenario 3A, 86,27%). Pola ini mengindikasikan bahwa pada arsitektur VGG16 dengan jumlah dataset sebesar 1.024 citra, rasio 75:15:10 merupakan titik keseimbangan optimal antara jumlah data latih yang cukup untuk pembelajaran model dan jumlah data validasi yang memadai untuk mengarahkan proses optimasi secara efektif. Penambahan proporsi data latih lebih lanjut hingga 80% pada Skenario 3A justru menghasilkan

penurunan akurasi uji, yang dapat disebabkan oleh berkurangnya jumlah data validasi menjadi hanya 10% sehingga estimasi generalisasi model selama pelatihan menjadi kurang konsisten, sebagaimana juga terlihat dari fluktuasi *validation loss* yang lebih besar pada grafik *training history* Skenario 3A. Meskipun demikian, Skenario 3A menunjukkan keunggulan pada aspek keseimbangan *precision-recall* kelas *Lumpy Skin* yang mencapai nilai seimbang pada angka 0,781, berbeda dengan Skenario 2A yang memiliki *precision* sempurna 1,000 namun dengan *recall* yang lebih rendah sebesar 0,656, menandakan bahwa penambahan data latih hingga 80% mendorong model untuk lebih berani dalam memprediksi kelas *Lumpy Skin* meskipun dengan konsekuensi munculnya beberapa *false positive*.

Pada kelompok skenario dengan arsitektur VGG19, pola yang sedikit berbeda dapat diamati. Skenario 1B (70:20:10) menghasilkan akurasi data uji terendah di antara seluruh skenario VGG19 sebesar 83,33%, kemudian meningkat pada Skenario 2B (75:15:10) menjadi 88,24%, dan bertahan pada angka yang sama untuk Skenario 3B (80:10:10) sebesar 88,24%. Lonjakan akurasi yang signifikan dari Skenario 1B ke Skenario 2B menunjukkan bahwa arsitektur VGG19 sangat diuntungkan oleh penambahan proporsi data latih dari 70% menjadi 75%, namun penambahan lebih lanjut ke 80% tidak lagi memberikan peningkatan akurasi yang berarti. Kondisi ini mengindikasikan bahwa arsitektur VGG19 yang lebih dalam memiliki ambang kebutuhan data latih minimum yang lebih tinggi dibandingkan VGG16, namun setelah ambang tersebut terpenuhi, penambahan data latih lebih lanjut tidak serta-merta meningkatkan performa secara proporsional. Secara keseluruhan, hasil penelitian ini menunjukkan bahwa dalam konteks dataset

berskala terbatas seperti yang digunakan dalam penelitian ini, pemilihan proporsi pembagian data perlu dilakukan secara cermat dengan mempertimbangkan karakteristik arsitektur model yang digunakan, karena rasio yang optimal untuk satu arsitektur belum tentu optimal untuk arsitektur lainnya.

4.4.2 Perbandingan Arsitektur VGG16 dan VGG19

Perbandingan antara arsitektur VGG16 dan VGG19 pada ketiga rasio pembagian data yang sama menghasilkan temuan yang menarik dan tidak sepenuhnya sesuai dengan asumsi umum bahwa model yang lebih dalam selalu menghasilkan performa yang lebih baik. Pada rasio 70:20:10, VGG16 (Skenario 1A) secara konsisten mengungguli VGG19 (Skenario 1B) pada seluruh metrik utama, dengan akurasi data uji 87,25% berbanding 83,33%, dan F1-Score kelas *Lumpy Skin* 0,745 berbanding 0,667. Perbedaan yang cukup signifikan ini mengindikasikan bahwa dengan jumlah data latih yang lebih terbatas (70% dari 1.024 citra, atau sekitar 717 sampel), kapasitas VGG19 yang lebih besar justru menyebabkan model lebih rentan terhadap *overfitting*, sebagaimana juga terlihat dari fluktuasi *validation loss* yang lebih besar pada grafik *training history* Skenario 1B.

Pada rasio 75:15:10, perbedaan performa antara kedua arsitektur menjadi lebih kecil. VGG16 (Skenario 2A) masih unggul dalam hal akurasi data uji (89,22% vs 88,24%), namun VGG19 (Skenario 2B) menunjukkan keunggulan yang berarti pada nilai *recall* kelas *Lumpy Skin* (0,781 vs 0,656) dan F1-Score kelas *Lumpy Skin* (0,806 vs 0,792). Kondisi ini menunjukkan bahwa dengan data latih yang lebih banyak, VGG19 mulai mampu memanfaatkan kapasitasnya yang lebih besar untuk

mempelajari representasi fitur kelas *Lumpy Skin* secara lebih komprehensif, meskipun belum sepenuhnya mengalahkan VGG16 dalam hal akurasi keseluruhan.

Pada rasio 80:10:10, VGG19 (Skenario 3B) berhasil mengungguli VGG16 (Skenario 3A) dalam hal akurasi data uji (88,24% vs 86,27%) dan nilai *precision* kelas *Lumpy Skin* (0,885 vs 0,781), namun VGG16 masih unggul pada nilai *recall* kelas *Lumpy Skin* (0,781 vs 0,719). Pola ini mengonfirmasi bahwa semakin besar proporsi data latih yang tersedia, semakin kecil kesenjangan performa antara kedua arsitektur, dan VGG19 cenderung menghasilkan akurasi keseluruhan yang lebih baik sementara VGG16 cenderung mempertahankan kemampuan deteksi kelas minoritas yang lebih konsisten.

Secara keseluruhan, temuan penelitian ini mengindikasikan bahwa dalam konteks dataset dengan jumlah citra yang terbatas seperti pada penelitian ini, arsitektur yang lebih sederhana seperti VGG16 tidak kalah kompetitif dibandingkan arsitektur yang lebih dalam seperti VGG19. Hal ini sejalan dengan prinsip umum dalam *deep learning* bahwa kompleksitas model perlu disesuaikan dengan skala dan karakteristik dataset yang tersedia. Penggunaan mekanisme *transfer learning* dari bobot ImageNet pada kedua arsitektur terbukti efektif dalam mengatasi keterbatasan jumlah data, memungkinkan kedua model untuk mencapai akurasi di atas 83% bahkan pada skenario dengan kondisi paling menantang sekalipun. Dengan demikian, pemilihan arsitektur yang tepat dalam kombinasi dengan proporsi pembagian data yang sesuai menjadi faktor krusial dalam membangun model klasifikasi *Lumpy Skin Disease* yang optimal pada dataset berskala terbatas, dan hasil penelitian ini menunjukkan bahwa VGG16 dengan rasio pembagian data

75:15:10 merupakan kombinasi terbaik yang menghasilkan performa paling optimal secara keseluruhan.

4.5 Integrasi Islam

Penelitian di bidang Teknik Informatika tidak seharusnya hanya berhenti pada pencapaian angka akurasi semata. Sebagai seorang Muslim dan khalifah di muka bumi, setiap inovasi teknologi yang dikembangkan sudah sepatutnya diarahkan untuk memberikan manfaat nyata bagi sesama dan merawat segala sesuatu yang telah diamanahkan Allah SWT kepada manusia, termasuk hewan ternak.

Sapi sebagai hewan ternak memegang posisi strategis dalam pemenuhan *maqashid syariah* (tujuan syariat), khususnya dalam hal *hifzh an-nafs* (menjaga kehidupan) melalui ketersediaan nutrisi, dan *hifzh al-maal* (menjaga harta) sebagai komoditas ekonomi masyarakat pedesaan. Urgensi penciptaan hewan ternak ini telah difirmankan secara lugas oleh Allah SWT dalam QS. An-Nahl ayat 5. Imam Ibnu Katsir dalam *Tafsir al-Qur'an al-'Azhim* menjelaskan bahwa kata *manāfi'* dalam ayat ini bersifat umum, mencakup seluruh kegunaan hewan ternak mulai dari susu, daging, hingga tenaganya (Ghoffar & Mu'thi, 1994). Ini mengisyaratkan kewajiban manusia untuk menjaga kesehatan hewan ternak sebagai bagian dari amanah kekhalifahan. Ketika sapi terserang LSD, manfaat tersebut akan berkurang bahkan hilang, sehingga upaya deteksi dini menjadi bentuk nyata penjagaan amanah tersebut.

Upaya ini sejalan dengan sabda Rasulullah SAW:

مَا أُنْزِلَ إِلَيْكَ مِنْ دَاءٍ إِلَّا أُنْزِلَ لَهُ شِفَاءٌ

"Tidaklah Allah menurunkan penyakit kecuali Dia juga menurunkan penawarnya."

(HR. Bukhari, No. 5678).

Imam Ibnu Hajar al-Asqalani dalam *Fath al-Bari* menegaskan bahwa penawar tersebut tidak datang sendirinya, melainkan harus diusahakan secara aktif melalui penelitian dan pengerahan akal (Abdullah, n.d.). Pengembangan sistem deteksi berbasis *deep learning* dalam penelitian ini merupakan wujud nyata ikhtiar tersebut.

Pemanfaatan kecerdasan buatan ini juga mencerminkan semangat QS. Ali Imran ayat 190. Imam al-Qurthubi dalam *al-Jami' li Ahkam al-Qur'an* menjelaskan bahwa *ulul albab* adalah mereka yang mendayagunakan akalnya untuk mengambil manfaat dari ciptaan Allah SWT (Utsman, n.d.). Sementara Wahbah az-Zuhaili dalam *Tafsir al-Munir* menambahkan bahwa pengamatan ilmiah terhadap alam termasuk bentuk ibadah bila diniatkan untuk kemaslahatan umat (Zuhaili, 2013). Sistem CNN berbasis VGG16 yang mempelajari pola visual kulit sapi dapat dipandang sebagai bentuk *tafakkur* berbasis teknologi yang sejalan dengan spirit ayat ini.

Sistem ini juga diharapkan dapat membantu para peternak di daerah terpencil yang memiliki keterbatasan akses terhadap tenaga dokter hewan, sehingga manfaatnya dapat dirasakan secara luas oleh masyarakat. Dengan demikian, penelitian ini merupakan salah satu bentuk pengamalan ilmu yang bermanfaat (*al-'ilm al-nafi'*), di mana teknologi tidak hanya dikembangkan untuk kepentingan akademis semata, tetapi juga diarahkan untuk kemaslahatan umat secara nyata, sebagaimana Rasulullah SAW bersabda:

خَيْرُ النَّاسِ أَنْفَعُهُمْ لِلنَّاسِ

"Sebaik-baik manusia adalah yang paling bermanfaat bagi manusia lainnya." (HR.

Ahmad, No. 23536)

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, penerapan metode *Convolutional Neural Network* (CNN) dengan arsitektur VGG dalam mengklasifikasikan penyakit kulit sapi terbukti efektif. Penerapan dilakukan melalui pendekatan *transfer learning* menggunakan bobot pra-latih (*pretrained weights*) ImageNet, sehingga model mampu mengekstraksi fitur visual yang relevan dari citra kulit sapi secara hierarkis mulai dari fitur tingkat rendah seperti tepi dan tekstur hingga fitur tingkat tinggi yang merepresentasikan pola lesi khas *Lumpy Skin Disease*, meskipun dataset yang digunakan relatif terbatas yaitu 1.024 citra yang terbagi ke dalam dua kelas (*Lumpy Skin* dan *Normal Skin*) dengan distribusi yang tidak seimbang antara kelas *Lumpy Skin* (324 citra) dan *Normal Skin* (700 citra).

Pengujian dilakukan melalui enam skenario menggunakan dua arsitektur VGG (VGG16 dan VGG19) dengan variasi proporsi pembagian data 70:20:10, 75:15:10, dan 80:10:10, menggunakan konfigurasi parameter yang sama. Seluruh skenario menghasilkan akurasi data uji di atas 83%, membuktikan bahwa kombinasi CNN-VGG dan *transfer learning* merupakan pendekatan yang andal untuk tugas klasifikasi penyakit kulit sapi meskipun dalam kondisi keterbatasan data. Perbandingan antara kedua arsitektur menunjukkan bahwa VGG16 secara umum memberikan performa yang lebih stabil pada dataset berskala terbatas, sementara VGG19 menunjukkan performa yang semakin kompetitif seiring

bertambahnya proporsi data latih dengan kesenjangan performa yang semakin mengecil pada rasio 75:15:10 dan 80:10:10. Proporsi pembagian data juga memberikan pengaruh yang berbeda terhadap masing-masing arsitektur, di mana pada VGG16 rasio 75:15:10 merupakan titik keseimbangan optimal yang menghasilkan akurasi tertinggi, sementara pada VGG19 penambahan proporsi data latih dari 70% ke 75% memberikan lonjakan performa yang signifikan namun tidak berlanjut secara proporsional pada rasio 80:10:10.

Performa terbaik dicapai oleh Skenario 2A yang menerapkan arsitektur VGG16 dengan proporsi pembagian data 75:15:10. Model terbaik ini selanjutnya berhasil diimplementasikan ke dalam aplikasi web berbasis Gradio dan mampu mengklasifikasikan 10 citra uji baru dengan tingkat keberhasilan 100%, membuktikan bahwa CNN dengan arsitektur VGG tidak hanya unggul secara metrik evaluasi tetapi juga dapat dioperasikan secara nyata sebagai sistem klasifikasi *Lumpy Skin Disease* pada sapi yang praktis dan andal.

5.2 Saran

Meskipun penelitian ini telah berhasil mencapai tujuannya dengan tingkat akurasi yang memuaskan, masih terdapat beberapa batasan yang dapat disempurnakan. Oleh karena itu, penulis merumuskan beberapa saran bagi peneliti selanjutnya maupun bagi pengguna hasil penelitian ini:

1. Pengembangan Dataset: Disarankan untuk memperbanyak kuantitas dan variasi dataset citra kulit sapi (seperti perbedaan pencahayaan, sudut pengambilan gambar dari jarak yang lebih jauh, serta keberagaman latar belakang kandang atau padang rumput). Penambahan data kelas penyakit

kulit sapi lainnya (seperti *Scabies* atau *Ringworm*) juga sangat direkomendasikan agar sistem mampu mendiagnosis berbagai jenis penyakit secara multikelas (*multiclass classification*).

2. Penggunaan Arsitektur yang Lebih Ringan: Untuk penelitian selanjutnya, disarankan membandingkan VGG dengan arsitektur CNN yang dirancang khusus untuk komputasi ringan (*mobile-friendly*), seperti *MobileNetV2* atau *DenseNet*. Hal ini bertujuan untuk mencari keseimbangan terbaik antara akurasi yang tinggi dengan ukuran model yang sangat kecil.
3. Pengembangan ke Platform Mobile (Android/iOS): Mengingat target pengguna akhir dari sistem ini adalah para peternak sapi dan petugas penyuluh lapangan (PPL) dokter hewan yang lebih sering beraktivitas di luar ruangan, disarankan agar penelitian ini dikembangkan menjadi aplikasi berbasis *mobile* (*smartphone*) terpadu. Dengan demikian, proses deteksi dini LSD dapat dilakukan dengan memotret sapi secara langsung di kandang tanpa bergantung penuh pada komputer atau laptop.

DAFTAR PUSTAKA

- Abdullah, S. A. A. (n.d.). *Fahthul Baari*. Pustaka Azzam.
- Ambilo, A., & Kebede, A. (2019). *Prevalence of major skin diseases of cattle in and*. 116–122. <https://doi.org/10.34297/AJBSR.2019.02.000586>
- Arnita, Marpaung, F., Aulia, F., Suryani, N., & Nabila, R. C. (2022). *COMPUTER VISION DAN PENGOLAHAN CITRA DIGITAL*. Pustaka Aksara.
- Bansal, M., Kumar, M., Sachdeva, M., & Mittal, A. (2023). Transfer learning for image classification using VGG19: Caltech-101 image data set. *Journal of Ambient Intelligence and Humanized Computing*, 14(4), 3609–3620. <https://doi.org/10.1007/s12652-021-03488-z>
- Bettencourt, E., & Narciso, V. (2015). *The Livestock Roles in the Wellbeing of Rural Communities of Timor-Leste The Livestock Roles in the Wellbeing of Rural Communities of Timor-Leste*. March. <https://doi.org/10.1590/1234-56781806-94790053s01005>
- BPS. (2024). *Populasi Ternak Menurut Provinsi dan Jenis Ternak (ekor)*, 2022. Badan Pusat Statistik. <https://www.bps.go.id/>
- Cahya, F. N., Hardi, N., Riana, D., & Hadiani, S. (2021). *Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network (CNN)*. 10, 618–626.
- Ermatita, Putra, E. D., Utami, M., R, M. A., Nugraha, W., & Noprisson, H. (2025). *Kecerdasan Artifisial Untuk Citra Digital*. CV Jejak.
- Fhadilla, N. (2024). *Implementasi Model Convolutional Neural Network Dengan Fitur Handcrafted Untuk Mendeteksi Perilaku Pengemudi* [UNiversitas Sebelas Maret]. <https://digilib.uns.ac.id/dokumen/detail/111545/Implementasi-Model-Convolutional-Neural-Network-Dengan-Fitur-Handcrafted-Untuk-Mendeteksi-Perilaku-Pengemudi>
- Ghoffar, M. A., & Mu'thi, A. (1994). *Tafsir Ibnu Katsir Jilid 5 (I)*. Pustaka Imam asy-Syafi'i.
- Gloriabarus. (2023). *Kekurangan Pakan Ternak Jadi Masalah Serius, Akibat Faktor Musim dan Pemanfaatan Lahan*. <https://ugm.ac.id/id/berita/kekurangan-pakan-ternak-jadi-masalah-serius-akibat-faktor-musim-dan-pemanfaatan-lahan/>
- Juandhi, M. D., Kurnia, D., & Anwar, P. (2019). *PENDUGAAN BODY CONDITION SCORING (BCS) TERHADAP BOBOT BADAN, BOBOT*

KARKAS DAN PERSENTASE KARKAS SAPI BRAHMAN CROSS (BX)
DI RPH KOTA PEKANBARU. *Journal of Animal Center*, 1(1), 37–45.

- Khani, N. I., & Rakasiwi, S. (2025). Penerapan Convolutional Neural Network dengan ResNet-50 untuk Klasifikasi Penyakit Kulit Wajah Efektif. *Edumatic: Jurnal Pendidikan Informatika*, 9(1), 217–225. <https://doi.org/10.29408/edumatic.v9i1.29572>
- Kusumo, D., Priyanti, A., & Saptati, R. A. (2007). *Prospek Pengembangan Usaha Peternakan Pola Integrasi*. 5(September), 26–33.
- Likhar, K., & Ridhorkar, S. (2024). Enhancing Skin Cancer Detection: A Comparative Analysis of Models with VGG-16, VGG-19, and Inception V3. *International Journal of Intelligent Systems and Applications in Engineering*, 12(10s), 502–514.
- Lojkic, I., Simic, I., Kresic, N., & Bedekovic, T. (2018). Complete Genome Sequence of a Lumpy Skin Disease Virus Strain Isolated from the Skin of a Vaccinated Animal. *Genome Announcements*, 6(22).
- Lubis, M. F., Hadinata, W., & Syahputra, G. (2025). *Analisis Perkembangan Populasi Dan Produktivitas Ternak Sapi Di Indonesia*. 1.
- Lum, K. Y., Goh, Y. H., & Lee, Y. Bin. (2020). American sign language recognition based on MobileNetV2. *Advances in Science, Technology and Engineering Systems*, 5(6), 481–488. <https://doi.org/10.25046/aj050657>
- Luthfi, N., Susanti, I., Nuraliah, S., Faradila, S., Suryani, H. F., Salido, W. L., Armayanti, A. K., Jannah, R., Khaeruddin, & Prima, A. (2024). *Pengantar Peternakan*. PT. Sonpedia Publishing Indonesia.
- Mahbub, L. A., Mulyani, E. D. S., & Ramadhan, T. I. (2025). Optimisasi Model Convolutional Neural Network Untuk Klasifikasi Penyakit Lumpy Skin Pada Sapi. *Jurnal Elektro & Infomatika Swadharma (JEIS)*, 05, 60–71.
- Nurtini, S., & UM, M. A. (2014). *Profil Peternakan Sapi Perah Rakyat di Indonesia*. Gajah Mada University Press.
- Pertanian, K. (2023). *Buku Outlook Komoditas Peternakan Daging Sapi*. Pusat Data dan Sistem Informasi Pertanian Sekretariat Jenderal - Kementerian Pertanian.
- Peryanto, A., Studi, P., Elektro, T., Dahlan, U. A., Studi, P., Teknik, M., & Dahlan, U. A. (2019). *Rancang Bangun Klasifikasi Citra Dengan Teknologi Deep Learning Berbasis Metode Convolutional Neural Network*. 8, 138–147.
- Pigou, L., Dieleman, S., Kindermans, P.-J., & Schrauwen, B. (2015). *Sign Language Recognition Using Convolutional Neural Networks* (pp. 572–578).

https://doi.org/10.1007/978-3-319-16178-5_40

- Putra, D. P., & Wiriasto, G. W. (2024). *Deteksi Penyakit Lumpy Hewan Ternak Menggunakan Metode Arsitektur MobileNetV2 Detection of Lumpy Disease in Livestock Using the MobileNetV2 Architecture Method*. 6(2), 149–162. <https://doi.org/10.30812/bite.v6i2.4401>
- Ravi, R., & Yadhukrishna, S. V. (2020). A Face Expression Recognition Using CNN & LBP. *Fourth International Conference on Computing Methodologies and Communication (ICCMC 2020)*, Iccmc, 684–689. <https://doi.org/10.1109/ICCMC48092.2020.ICCMC-000127>
- Safiq Alfiansyah, N., & Litanianda, Y. (2024). Identifikasi Lumpy Skin Disease Menggunakan Tensorflow Dengan Metode Convolutional Neuron Network. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(4), 7330–7336. <https://doi.org/10.36040/jati.v8i4.10238>
- Sari, N. P. (n.d.). *Analisis Performa Algoritma CNN dalam Klasifikasi Citra Medis Berbasis Deep Learning Analysis Of CNN Algorithm In Deep Learning-Based Medical Image Classification*. 2(2), 87–92.
- Sendow, I., Assadah, N. S., Ratnawati, A., Dharmayanti, N., & Saepulloh, M. (2021). *Lumpy Skin Disease : Ancaman Penyakit Emerging bagi Status Kesehatan Hewan Nasional (Lumpy Skin Disease : Emerging Diseases Threats for National Animal Health Status)*. 31(2), 85–96.
- Sentoso, T., Ardiansyah, F., Tamuntuan, V., Sastra Wangsa, S., Kusrini, & Kusnawi. (2024). Identifikasi Lumpy Skin Disease pada Ternak Sapi dengan Klasifikasi Citra menggunakan Metode Convolutional Neural Network. *Sistemasi: Jurnal Sistem Informasi*, 13(3), 864–873. <https://sistemasi.ftik.unisi.ac.id/index.php/stmsi/article/view/2569/691>
- Shidik, F. A., Mustofa, K., Kartiningtyas, A. P., & Agustin, T. (2024). Analisis Citra Medis Untuk Identifikasi Penyakit Mata Dengan Teknologi Convolutional Neural Networks. *Seminar Nasional Amikom Surakarta (SEMNAS) 2024, November*, 68–80.
- Sugihandono, A., & Fatta, H. Al. (2019). *Case Base Reasoning Sebagai Alat Bantu Diagnosis Penyakit Ternak Sapi menggunakan Sorenson Coeficient (Studi Kasus : Di Kabupaten Pati)*. XIV, 31–42.
- Sulistyo, W. (2025). *Peran Peternakan Digital dalam Mendukung Efisiensi Produksi Ternak*. 4(April).
- Tawaf, R. (2018). ANALISIS USAHA PEMBIAKAN SAPI POTONG POLA KEMITRAAN ANTARA KORPORASI DENGAN PETERNAK RAKYAT. *Jurnal Ilmu-Ilmu Sosial Dan Humaniora*, 20(1), 45–56.

Utsman, M. H. (n.d.). *Tafsir Al Qurthubi*. Pustaka Azzam.

Wardani, K. R. R., Suryalim, H., Engel, V. J. L., & Christian, H. (2023). Analisis Pemilihan Optimizer dalam Arsitektur Convolution Neural Network VGG16 dan Inception untuk Sistem Pengenalan Wajah. *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, 9(2), 186. <https://doi.org/10.26418/jp.v9i2.60432>

Wona, M. M. A., Rahayu, W., Wirantasa, U., Studi, P., Informatika, T., Gedong, K., Rebo, P., & Timur, J. (2025). *Klasifikasi dan deteksi penyakit kulit melalui pengolahan citra dengan metode cnn*. 06(01), 43–51.

Yang, W., Zhang, D., & Fu, Y. (2016). Research of a Diagonal Recurrent Neural Network and Artificial Neural Networks. *2016 International Symposium on Computer, Consumer and Control (IS3C)*, 374–377. <https://doi.org/10.1109/IS3C.2016.103>

Zuhaili, W. (2013). *Tafsir Al-Munir*. Gema Insani.